



## Calhoun: The NPS Institutional Archive

---

Theses and Dissertations

Thesis Collection

---

2012-09

# Analysis of OPACITY and PLAID Protocols for Contactless Smart Cards

Kiat, Koh Ho

Monterey, California. Naval Postgraduate School

---

<http://hdl.handle.net/10945/17385>



Calhoun is a project of the Dudley Knox Library at NPS, furthering the precepts and goals of open government and government transparency. All information contained herein has been approved for release by the NPS Public Affairs Officer.

**Dudley Knox Library / Naval Postgraduate School**  
**411 Dyer Road / 1 University Circle**  
**Monterey, California USA 93943**

<http://www.nps.edu/library>



# **NAVAL POSTGRADUATE SCHOOL**

**MONTEREY, CALIFORNIA**

## **THESIS**

**ANALYSIS OF OPACITY AND PLAID PROTOCOLS  
FOR CONTACTLESS SMART CARDS**

by

Koh Ho Kiat  
Lee Yong Run

September 2012

Thesis Co-Advisors:

John D. Fulp  
Gurminder Singh

**Approved for public release; distribution is unlimited**

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.				
1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE September 2012		3. REPORT TYPE AND DATES COVERED Master's Thesis
4. TITLE AND SUBTITLE Analysis of OPACITY and PLAID Protocols for Contactless Smart Cards			5. FUNDING NUMBERS	
6. AUTHOR(S) Koh Ho Kiat and Lee Yong Run				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000			8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A			10. SPONSORING/MONITORING AGENCY REPORT NUMBER	
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. IRB Protocol number N/A.				
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited			12b. DISTRIBUTION CODE A	
13. ABSTRACT (maximum 200 words)  There is an increasing use of contactless smart card technology for identification, access control, and financial transactions due to its numerous advantages. However, there is also an increasing number of attacks that exploit the insecure contactless communications in order to gain unauthorized access to personal and sensitive information for illegitimate use. Open Protocol for Access Control Identification and Ticketing with privacy (OPACITY) and Protocol for Lightweight Authentication of Identity (PLAID) are two privacy-enhanced protocols that enable secure contactless communications to protect the confidentiality, integrity, and authenticity of contactless smart card information and transactions.  This thesis will examine and analyze the principle mechanisms behind OPACITY and PLAID protocols to determine the strengths and weaknesses of the protocols, as well as to benchmark the performance of the protocols against each other.				
14. SUBJECT TERMS Open Protocol for Access Control Identification and Ticketing with Privacy, OPACITY, Protocol for Lightweight Authentication of Identity, PLAID, Wireless authentication, Contactless smart cards authentication, factors of authentication, granularity of identification, credential confidence, subject-token binding, non-repudiation, MITM resistance, protection of classified material, authorization support, key distribution, bit entropy, cipher performance			15. NUMBER OF PAGES 114	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UU	

NSN 7540-01-280-5500

Standard Form 298 (Rev. 8-98)  
Prescribed by ANSI Std. Z39.18

THIS PAGE INTENTIONALLY LEFT BLANK

**Approved for public released; distribution is unlimited**

**ANALYSIS OF OPACITY AND PLAID PROTOCOLS FOR CONTACTLESS  
SMART CARDS**

Koh Ho Kiat  
Civilian, Defence Science and Technology Agency (DSTA), Singapore  
B.Eng., National University of Singapore, 2007

Lee Yong Run  
Captain, Singapore Armed Forces (SAF)  
B.Sc., University of Melbourne, 2008

Submitted in partial fulfillment of the  
requirements for the degree of

**MASTERS OF SCIENCE IN COMPUTER SCIENCE  
(INFORMATION ASSURANCE)**

from the

**NAVAL POSTGRADUATE SCHOOL  
September 2012**

Author: Koh Ho Kiat  
Lee Yong Run

Approved by: John D. Fulp  
Co-Advisor

Dr. Gurminder Singh  
Co-Advisor

Dr. Peter Denning  
Chair, Department of Computer Science

THIS PAGE INTENTIONALLY LEFT BLANK

## **ABSTRACT**

There is an increasing use of contactless smart card technology for identification, access control, and financial transactions due to its numerous advantages. However, there is also an increasing number of attacks that exploit the insecure contactless communications in order to gain unauthorized access to personal and sensitive information for illegitimate use. The Open Protocol for Access Control Identification and Ticketing with privacy (OPACITY) and the Protocol for Lightweight Authentication of Identity (PLAID) are two privacy-enhanced protocols that enable secure contactless communications to protect the confidentiality, integrity, and authenticity of contactless smart card information and transactions.

This thesis will examine and analyze the principle mechanisms behind the OPACITY and the PLAID protocols to determine the strengths and weaknesses of the protocols, as well as to benchmark the performance of the protocols against each other.



THIS PAGE INTENTIONALLY LEFT BLANK

## TABLE OF CONTENTS

I.	INTRODUCTION.....	1
A.	INTRODUCTION.....	1
B.	MOTIVATION.....	2
C.	RESEARCH METHOD.....	3
D.	THESIS ORGANIZATION.....	4
II.	OVERVIEW OF CONTACTLESS SMART CARD TECHNOLOGY.....	5
A.	HISTORY OF SMART CARDS.....	5
B.	ARCHITECTURE.....	5
C.	STANDARDS.....	6
	1. ISO 7816.....	7
	2. ISO 14443.....	7
	3. ISO 24727.....	7
D.	APPLICATIONS.....	7
	1. Identification.....	8
	2. Financial.....	9
	3. Access Control.....	9
E.	THREATS AND VULNERABILITIES.....	10
	1. Eavesdropping.....	10
	2. Impersonation.....	10
	3. Denial of Service (DOS).....	11
	4. Relay Attacks.....	12
	5. Radio Frequency Analysis.....	13
F.	SECURITY REQUIREMENTS.....	13
	1. Confidentiality.....	14
	2. Integrity.....	14
	3. Availability.....	15
	4. Non-repudiation.....	15
III.	OPACITY.....	17
A.	OVERVIEW.....	17
B.	STANDARDS / RECOMMENDATIONS.....	18
C.	GENERAL CHARACTERISTICS.....	18
	1. Single Factor Authentication Protocol.....	18
	2. Asymmetric-Based Authentication.....	19
	3. Authentication versus Authorization.....	20
D.	FEATURES.....	21
	1. Security Features.....	21
	a. Authentication.....	21
	b. End-to-End Protection with Forward Secrecy.....	22
	c. Identity Privacy.....	22
	2. Performance Features.....	23
	a. Persistent Binding.....	23

	b.	<i>Anti-tearing and Synchronization</i> .....	23
	c.	<i>Simple Integration and Interoperability</i> .....	24
E.		MODES OF OPERATION .....	24
	1.	OPACITY-FS .....	24
	2.	OPACITY-ZKM .....	25
F.		CIPHER SUITES .....	25
G.		OPACITY-FS AUTHENTICATION .....	26
	1.	Step 0: Preloading of Required Authentication Information .....	27
	2.	Step1: Card Terminal (Host) Initiates Authentication Request .....	28
	3.	Step 2: Smart Card (ICC) Generates Authentication Cryptogram .....	29
	a.	<i>Validates Host CVC</i> .....	29
	b.	<i>Verifies Persistent Binding Feature</i> .....	29
	c.	<i>Derives Session Keys and Generates Authentication Cryptogram</i> .....	32
	3.	Step 3: Card Terminal (Host) Verifies Authentication Cryptogram .....	34
	a.	<i>Verifies Persistent Binding Feature</i> .....	34
	b.	<i>Derives Session Keys and Verifies Authentication Cryptogram</i> .....	36
H.		STRENGTHS AND LIMITATIONS.....	38
	1.	Strengths .....	38
	a.	<i>Asymmetric Cryptography</i> .....	38
	b.	<i>Use of Different Session Keys</i> .....	39
	c.	<i>Mutual Authentication and End-end Protection with Forward Secrecy and Identity Privacy</i> .....	39
	2.	Limitations.....	39
	a.	<i>Asymmetric Authenticating</i> .....	39
	b.	<i>Single Factor Authentication</i> .....	40
	c.	<i>Bearer token</i> .....	40
IV.		PLAID .....	43
	A.	OVERVIEW .....	43
	B.	STANDARDS/RECOMMENDATIONS.....	43
	C.	GENERAL CHARACTERISTICS.....	44
	1.	Multi-Factor Authentication .....	44
	2.	Symmetric and Asymmetric Key Algorithms .....	44
	3.	Authentication versus Authorization .....	45
	D.	FEATURES .....	46
	1.	Security Features.....	46
	a.	<i>Mutual Authentication</i> .....	46
	b.	<i>End-to-End Protection</i> .....	46
	c.	<i>Authorization Checks</i> .....	47
	2.	Performance Features .....	47

	a.	<i>Simple Integration and Interoperability</i> .....	47
E.		MODES OF OPERATIONS.....	47
F.		SUGGESTED KEY LENGTHS AND ALGORITHMS.....	48
G.		AUTHENTICATION PROCESS .....	48
	1.	Step 0: Preloading of Required Authentication Information .....	50
	2.	Step 1: IFD Initiates Authentication Command .....	51
	3.	Step 2: ICC Responds to the IA Command by Generating IA Response .....	51
	4.	Step 3: IFD Responds to the IA Response from ICC by Generating FA Command.....	52
	5.	Step 4: ICC Response to the FA Command.....	54
	6.	Step 5: IFD Processes Credentials.....	55
H.		STRENGTHS AND LIMITATIONS.....	55
	1.	Strengths .....	56
	a.	<i>Hybrid Cryptography</i> .....	56
	b.	<i>Multi-Factor Authentication</i> .....	57
	c.	<i>Mutual Authentication, End-To-End Protection</i> .....	57
	2.	Limitations.....	57
	a.	<i>Slow in Comparison to AES Only Authentication</i> ....	57
	b.	<i>Key Distribution Problem</i> .....	58
V.		METHODOLOGY.....	59
A.		INTRODUCTION .....	59
B.		DATA ANALYSIS .....	59
	1.	Open Coding .....	59
	2.	Axial Coding.....	60
	3.	Selective Coding.....	62
C.		STANDARDS AND GUIDELINES .....	63
	1.	Federal Information Processing Standards 140–2 Publication (FIPS Pub 140–2) .....	64
	2.	Federal Information Processing Standards 201–2 Publication (FIPS Pub 201–2) .....	66
	3.	NIST Special Publication 800–63–1: Electronic Authentication Guideline. ....	67
	4.	NIST Special Publication 800–116: A Recommendation for the Use of PIV Credentials in Physical Access Control Systems (PACS).....	69
	5.	NSA Suite B Cryptography website / NSA The Case for Elliptic Curve Cryptography .....	70
	6.	Payment Card Industry Data Security Standard (PCI-DSS) / ISO 7816–8 .....	71
VI.		COMPARISONS AND FINDINGS .....	73
A.		COMPARISONS .....	73
	1.	Factors of Authentication .....	75
	2.	Granularity of Identification .....	76

3.	Credential Confidence.....	76
4.	Subject-Token Binding.....	77
5.	Non-Repudiation.....	78
6.	Man-in-the-Middle Resistance .....	79
7.	Protection of Classified Material .....	79
8.	Authorization Support.....	80
9.	Key Management .....	81
10.	Bit Entropy .....	82
11.	Cipher Performance .....	82
B.	SUMMARY OF FINDINGS.....	83
VII.	CONCLUSIONS .....	85
	LIST OF REFERENCES.....	87
	INITIAL DISTRIBUTION LIST .....	91

## LIST OF FIGURES

Figure 1.	Basic System Setup for Relay Attack [18] (modified).....	12
Figure 2.	Setup of Smart Card and Card Terminal Using OPACITY Protocol [19] (modified).....	18
Figure 3.	The OPACITY Protocol General Authentication Process .....	27
Figure 4.	Preloaded Authentication Information .....	28
Figure 5.	Card Terminal Initiates Authentication Request [19] (modified).....	29
Figure 6.	Validate Host CVC [19] (modified) .....	29
Figure 7.	ICC PB Feature Check Case 1 and Case 2 [19] (modified) .....	31
Figure 8.	ICC PB Feature Check Case 3 [19] (modified) .....	32
Figure 9.	Derive Session Keys and Authentication Cryptogram [19] (modified) .....	33
Figure 10.	Host PB Feature Check Case 1 [19] (modified) .....	35
Figure 11.	Host PB Feature Check Case 2 [19] (modified) .....	36
Figure 12.	Host PB Feature Check Case 3 [19] (modified) .....	36
Figure 13.	Host Verifies Authentication Cryptogram [19] (modified) .....	37
Figure 14.	The PLAID Protocol General Authentication Process .....	50

THIS PAGE INTENTIONALLY LEFT BLANK

## LIST OF TABLES

Table 1.	Cipher Suites Supported By The OPACITY Protocol.....	25
Table 2.	Cipher Suites Supported By The PLAID Protocol.....	48
Table 3.	Compilation of Reviewed Standards.....	60
Table 4.	Compilation of Reviewed Standards.....	63
Table 5.	Authentication for Physical Access .....	66
Table 6.	Authentication for Logical Access .....	66
Table 7.	Assurance Levels for Multi-Token Authentication Schemes .....	68
Table 8.	Authentication Factors of PIV Authentication Mechanisms .....	69
Table 9.	Authentication Factors for Security Areas .....	70
Table 10.	Summary of Comparisons .....	74
Table 11.	NIST Recommended Key Sizes .....	82



THIS PAGE INTENTIONALLY LEFT BLANK

## **LIST OF ACRONYMS AND ABBREVIATIONS**

ACL	Access Control List
ACS	Access Control System
AES	Advanced Encryption Standard
APDU	Application Protocol Data Unit
ASN	Abstract Syntax Notation Number
ATM	Automatic Teller Machines
CBC	Cipher Block Chaining
CIA	Confidentiality, Integrity and Availability
CVC	Card Verifiable Certificate
DivData	Diversification Data
DOS	Denial of Service
ECDH	Elliptic Curve Diffie-Hellman
ECDSA	Elliptic Curve Digital Signature Algorithm
E-IDs	Electronic Identity Cards
EMV	Europay, Mastercard and Visa
E-Passports	Electronic Passports
ESTR	Encrypted String
FA	Final Authentication
FIPS	Federal Information Processing Standards
HIPAA	Health Insurance Portability and Accountability Act
IA	Initial Authentication

ICC	Integrated Circuit Card or Smart Card
IFD	Interface Device
ISO	International Organization for Standardization
Keyset ID	Key Set Identifier
KDF	Key Derivation Function
LACS	Logical Access Control System
MAC	Message Authentication Code
MITM	Man-In-The-Middle
NSA	National Security Agency
OPACITY privacy	Open Protocol for Access Control Identification and Ticketing with privacy
OPACITY-FS	OPACITY Full Secrecy
OPACITY-ZKM	OPACITY Zero Key Management
OpModeID	Operational Mode Identifier
PACS	Physical Access Control System
PB	Persistent Binding
PCI-DSS	Payment Card Industry Data Security Standard
PFS	Perfect Forward Secrecy
PII	Personal Identifiable Information
PIN	Personal Identification Number
PIV	Personal Identity Verification
PKC	Public Key Cryptography
PLAID	Protocol for Lightweight Authentication of IDentity

RND	Random Number
RNG	Random Number Generator
SAM	Secure Authentication Module
SHA	Secure Hash Algorithm
STR	String
TRNG	True Random Number Generator

THIS PAGE INTENTIONALLY LEFT BLANK

## **ACKNOWLEDGMENTS**

Ho Kiat would like to thank his employer, Defence Science and Technology Agency (DSTA), for the opportunity to continue his education. In addition, Ho Kiat would also like to thank his wife, Liu Qiulin, for her unwavering support, understanding, and unconditional love during the writing of this thesis. It was only through her support and personal sacrifice that this thesis became a reality.

Yong Run would also like to thank his employer, the Singapore Armed Forces (SAF), for providing him the opportunity to expand his horizons. Yong Run would like to thank his parents, brothers and friends who have given him their unequivocal support throughout, as always, for which a mere expression of thanks would not suffice. It would also not have been possible to write this thesis without the help and support of his partner, Ho Kiat.

Finally, both authors wish to express their sincere gratitude to their advisors Mr J.D. Fulp and Dr Gurminder Singh for their patience, motivation, enthusiasm and immense knowledge. Every meeting was a rush of ideas and every draft thrown back, spurred the authors to work harder. Lastly, the authors would like to express their gratitude to those who have contributed to the editing, reading and processing of this document. Your time, patience and invaluable advice are greatly appreciated. Thank you!

THIS PAGE INTENTIONALLY LEFT BLANK

## **I. INTRODUCTION**

### **A. INTRODUCTION**

With the proliferation of information technology, organizations today are beginning to realize that protection of business information from malicious (and non-malicious yet, nonetheless, harmful) acts is a growing concern [1]. The fact that information security is multi-faceted and complex results in many organizations struggling to implement an effective set of security practices [2]. In particular, organizations are focusing on effectively administering access to resources by employing physical access controls as well as logical access controls to protect valuable resources.

Traditionally, the means of coordinating people and privileges have always revolved around the need to establish identity. An example of this is the use of a library card to allow a library to determine how much and what types of books an individual is allowed to borrow. Other examples include driving licenses, credit cards, and employee identification cards. In the past, these cards relied on a trusted party to verify that the holder had the identity as represented on the card. The party then mapped the identity to determine the holder's rights and privileges. However, with the increased prevalence of technology, this old and easily subverted method has been replaced with newer technologies that enable greater confidence in the veracity of the identity presented. The newer technologies include electronic devices such as card readers that are employed to automate the identification process, resulting in reduced cost, increased convenience, and more trustworthy implementations [3].

Although such technologies offer significant benefits, there is, at the core of the process, a more complicated electronic transaction that must be thoroughly scrutinized to ensure that it delivers a verified identity and is not (at least easily) subvertible. Since all forms of access control—whether physical or logical—depend greatly on the accuracy of the subject identification process, getting this correct is of utmost importance. Additionally, to further elaborate on



the access control issues faced by organizations today, many companies see a frequent turnover of employees or contractors, which may allow unauthorized people to gain access to valuable information.

The introduction of contactless smart card systems greatly improves the usability of smart card systems. Users no longer need to carry multiple credentials to access different applications and access rights can be controlled from a central location to easily grant or withdraw access rights in a single transaction, reducing the complexity of the process and lowering the overall maintenance cost and effort [3]. In addition, there is also improved security as some of the smart card systems use cryptographic algorithms to enhance the security of the transactions.

While there is sufficient literature on how contactless smart cards should be used in organizations, less is known about the different security protocols employed by the cards. Since the introduction of contactless smart cards in 2009, two privacy-enhanced protocols, OPACITY (Open Protocol for Access Control Identification and Ticketing with privacy) [4] and PLAID (Protocol for Lightweight Authentication of ID) [5], have garnered little attention from the industry or academia. Furthermore, the two protocols have never been compared in order to determine their relative security strengths and limitations. This shortage of protocol scrutiny provides a fair justification for the need to examine and compare these two security protocols for contactless smart card applications, involving logical access control (LAC) and physical access control (PAC) transactions.

## **B. MOTIVATION**

Although most organizations have implemented the use of contactless smart cards for access control in accordance with international standards or recommended guidelines, many of the contactless smart cards have failed to achieve their desired effect, resulting in security lapses. This is because organizations do not fully understand how these protocols work. Typically, organizations do not select the optimal protocol based on the strengths and

limitations that best match their security policy, but instead they select the protocol based on vendor recommendations.

The need for research in this area is also justified from other perspectives. Firstly, this thesis aims to address the lack of available literature on OPACITY and PLAID protocols as described previously. This thesis examines and analyzes the principle security mechanisms behind OPACITY and PLAID protocols that enable them to provide secure transactions. It determines the strengths and limitations of the protocols, and compares the protocols to each other. The usefulness of this thesis will be evident to researchers and industry seeking to understand the differences between both protocols when considering which to adopt/employ.

Secondly, this thesis aims to determine how best to evaluate and compare the two different protocols using standards/guidelines, such as International Organization for Standardization (ISO) 24727 and Federal Processing Information Standards (FIPS) 140–2 as a baseline for comparative evaluation. We suggest a methodology where excerpts from different standards/guidelines are examined to create a list of factors that can be used for comparison between the two protocols. Firstly, we examine the requirements recommended or stipulated by relevant guidelines and standards to identify suitable factors that can be used for comparison. Secondly, we use the identified factors to compare the factors.

### **C. RESEARCH METHOD**

As the protocols are relatively new, very little research has been performed on them, resulting in limited literature on these protocols. Hence, the research method will comprise mainly of literature review, using materials released by the protocol developers and the industry on contactless smart card.

The use of literature review provides the theoretical framework for further planning and study. It is also a particularly effective approach for deriving the different criteria for comparison. Another benefit of literature review is that it allows us to familiarize ourselves with the protocols so that we can articulate the

key research issues better after performing a thorough comparative study. A high level of product and intrinsic computer security knowledge is required to ensure that the information written is accurate and meaningful.

#### **D. THESIS ORGANIZATION**

This thesis is broken into three sections and it consists of seven chapters.

Chapter I describes the motivation for this thesis and outlines the content of each chapter.

Chapter II lays the foundation for this thesis and provides an overview of contactless smart card systems that includes the architecture, standards, applications, as well as the threats, vulnerabilities, and security requirements of contactless smart card systems. The threats, vulnerabilities, and security requirements will drive the need for OPACITY and PLAID protocols, which is the focus of subsequent chapters and forms the core of this thesis.

Chapter III and Chapter IV describe the OPACITY and PLAID protocols, respectively. The chapters include an overview of the protocols, their features, and the corresponding principle security mechanisms that provide the necessary security for contactless smart card transactions. The next two chapters focus on the comparisons between the protocols.

Chapter V describes the methodology and it lists the factors that have been identified to compare the protocols, while Chapter 6 presents the results, analysis, and conclusions drawn from the comparisons.

Finally, Chapter VII provides recommendations for future work and summarize the thesis respectively.

## **II. OVERVIEW OF CONTACTLESS SMART CARD TECHNOLOGY**

### **A. HISTORY OF SMART CARDS**

The history of smart cards can be traced back to the early 1950s when the Diner's Club introduced the use of plastic cards that their customers could use for payment applications in the United States. This revolutionized the way in which customers could make payments as these cards provided the customer with a form of identity to a select group, and businesses that recognized this group would accept payments using these cards.

However, in the 1960s, the cost pressures of card frauds and card tampering necessitated a card that would overcome these challenges. Finally, in 1968, German inventors Jurgen Dethloff and Helmet Grotrupp filed a patent for using plastic as carriers for microchips, which resulted in the creation of the world's first smart cards [6]. These microchips have security mechanisms that make them an ideal medium for safely storing cryptographic keys and algorithms used to more assuredly identify a card's owner.

Banks subsequently began to use these types of cards and governments soon followed suit by issuing such cards to citizens as forms of identity. Today, such technology is prevalent in our daily lives.

### **B. ARCHITECTURE**

A smart card can essentially be defined as any form of pocket-sized card that is embedded with integrated circuits that allows it to process and store information. Essentially, there are two broad categories of smart cards, namely contact-based and contactless smart cards. However, for the purpose of this thesis, the primary focus is on the contactless smart cards.

Contactless smart card technology is often utilized in applications that are used to protect personal information or deliver secure transactions. In particular,

there are many different ways that contactless smart cards utilize embedded antennas to exchange data stored in the chip's memory to a remote device.

For the purpose of communications security, international standards limit the operating range of most contactless smart cards to approximately 3 to 4 inches [7]. However, applications that require longer reading distances can rely on other forms of contactless technologies, such as RFID. Contactless smart cards contain a re-writable smart card microchip that is used to read or write via radio waves. Depending on how these contactless smart cards are implemented, processing and storage of information may also be performed using a microprocessor found in the chip itself. Most modern contactless smart cards rely on a built-in inductor to capture incident electromagnetic energy that is then used to power the chip electronics. These cards provide significantly higher levels of security by implementing encryption [8], have a much larger memory storage capacity and are able to have information written onto them in real time, allowing a single card to be used for numerous applications, such as access control, vending, or fare collection.

### **C. STANDARDS**

This section provides a brief introduction to some of the standards used for contactless smart card systems. Standards are important as they often reduce the cost of technology adoption for organizations. Perhaps the most important purpose of standards is that they provide an amalgamation of the industry's best practices to ensure issues such as inter-operability and implementation can be resolved efficiently.

The ISO 7816, ISO 14443, and ISO 24727 series are the common standards for contactless smart card systems. It is important to know that these standards should not be implemented in isolation. Instead, these standards should be used to complement one another in order to produce a more robust

product. An example is the Europay, Mastercard and VISA (EMV) standard, which uses the ISO 7816 and ISO 14443 series to define the interface standards for financial transactions.

#### **1. ISO 7816**

One of the standards most commonly related to smart cards is the ISO 7816 series standard. This series consists of fifteen parts and defines details such as card dimensions, type of electronic signals, and transmission protocols.

#### **2. ISO 14443**

Similar to the ISO 7816 series, the ISO 14443 series is another international standard that describes how contactless smart cards and terminals should work to ensure industry-wide compatibility. The ISO 14443 series addresses the card's physical characteristics, radio frequency power, and signal interface, initialization, and anti-collision, as well as the transmission protocol of contactless smart card systems.

#### **3. ISO 24727**

In the area of security for smart cards, the ISO 24727 series is the first international standard that addresses the need for a layered framework to support interoperability of smart cards providing security features such as identification, authentication, and digital signatures.

### **D. APPLICATIONS**

There is an increasing use of contactless smart card systems due to the convenience, performance, and basic security that the systems provide, as well as the ease of integrating these systems for use in a wide range of applications.

Based on a market analysis report released by Frost & Sullivan in 2008 on the world outlook for deployment of contactless smart cards, the potential gross revenue was forecasted to reach approximately U.S.\$2 billion in 2012 based on a compound annual growth rate of approximately 21.7% from 2006 to 2012 [9]. In

another recent market analysis report released by Frost & Sullivan in 2010 on the contactless smart card deployment in the Asia-Pacific region, the region's gross revenue alone was forecasted to reach approximately U.S.\$2 billion by the close of 2016, with 1.9 billion contactless smart cards estimated to be shipped [10]. This is a large increase compared to the 590 million contactless smart cards that were shipped in 2009. A similar market analysis conducted by IMS Research also forecasted that there would be an increasing demand for contactless smart cards [11]. The analysis predicted that the world's contactless smart card shipment would increase from 950 million in 2010 to 3.5 billion in 2016.

These reports are largely in agreement that the use of contactless smart cards as a means for identifications, financial applications, and access controls are the main drivers for the rise in the use of contactless smart cards.

## **1. Identification**

The primary market for the use of contactless smart cards as a means of identification is for government identification credentials such as electronic passports (e-passports) and electronic ID cards (e-IDs). The e-passports and e-IDs have a small contactless chip (i.e., integrated circuit) embedded in each of them. The chip is used to store personally identifiable information (PII) which is used to identify the identity of the card's owner. In addition, the cards are designed with multiple layers of security, such as cryptography and multi-factor authentication, to protect the PII from identity theft. These documents are less prone to identity theft compared to traditional identification documents.

There is also an increasing use of contactless smart cards for healthcare identification purposes. The smart cards enable administrative efficiencies in addition to higher assurance identification. The cards facilitate automated management of medical records (e.g., automated filing and retrieving of medical records) and mitigate human errors in managing the records. The cards are designed to comply with the Health Insurance Portable and Accountability Act (HIPAA) [12].

## **2. Financial**

The use of contactless smart cards for transportation payments was one of the first financial applications of contactless smart cards. This application has since been adopted by many countries such as the United States, Hong Kong, and Singapore. The Hong Kong Octopus card launched in 1997 as an electronic purse for public transportation is the most successful and mature implementation of contactless smart cards for mass transit payments [13]. Today, the Octopus card is also used for making payments at supermarkets and vending machines, and for street parking and admission to certain public facilities.

Many banks are also replacing contact-based credit cards and debit cards with contactless credit cards and debit cards. These contactless cards are in compliance with the EMV standard and are commonly known as “chip and pin” cards [14]. When a customer wants to make a payment, the card terminal will first verify the authenticity of the card using the secret key residing in the chip. Once the authenticity is verified, the customer is required to enter a personal identification number (PIN), which will be sent to the chip for verification. If the pin is correct, the chip will inform the card terminal and the card terminal will approve the transaction. If the authenticity is not verified or the pin is incorrect, the transaction is denied.

## **3. Access Control**

Access control is one of the most common applications of contactless smart cards. In these cases, contactless smart cards are used to store identification information and associated access privileges to control access to different locations. For example, contactless smart cards are used by hotels to control the access of hotel guests to different facilities in the hotel. Average guests may only be able to access their own room while privileged guests may have additional accesses to the gymnasium and business lounge. For more stringent access control to sensitive or classified premises, the cards can also store biometric information to further strengthen the authentication process



## **E. THREATS AND VULNERABILITIES**

Even though contactless smart cards provide considerable convenience to the users because they do not require the users to manually insert smart cards into the card terminals, there are inherent threats to the contactless feature (i.e., over the air communication between the smart cards and the card terminals) and vulnerabilities in the systems that can be exploited by attackers.

Examples of threats include eavesdropping, impersonation, denial of service (DOS), replay, and radio frequency analysis attacks. Examples of vulnerabilities include unprotected communications, use of protocols with weak security features, and poor system design and/or implementation.

### **1. Eavesdropping**

Eavesdropping is the simplest way to attack a contactless smart card. Even though the nominal operating range between some contactless smart cards and card terminals may only be 10 cm, the attacker is still able to listen passively to the over the air communications from a distance using high-gain directional antennas [15]. Using this form of attack, the attacker can capture sensitive personal information (e.g., biometric information) that is exchanged between the smart cards and the card terminals.

Eavesdropping is an attack on the confidentiality of the information and is effective on systems where information is exchanged in the clear (i.e., not encrypted) which the attacker can easily capture and read. If the information is encrypted before sending, the attacker will require more sophisticated cryptanalytic or key discovery techniques to decrypt the ciphertext. If strong ciphers and sound key management are employed, the attack may be completely thwarted.

### **2. Impersonation**

Impersonation, in the form of covert transaction attacks, is the biggest threat to contactless smart card systems as the user is unsure whether he is

interacting with a legitimate card terminal since the user is not required to interact physically with the card terminal [16]. Even if the user is interacting physically with the card terminal, the user may not be able to identify the differences between a genuine and a well-fabricated counterfeit card terminal.

There are many types of covert transactions. Examples include fraudulent merchants using illicit card terminals to interact with genuine cards to process unauthorized transactions, or fraudulent merchants processing multiple transactions concurrently. This attack has been used many times by the installation of fake Automatic Teller Machines (ATMs) that result in the unintended exposure of ATM cardholder authentication credentials (e.g., PIN).

These covert transactions are attacks on the integrity of the system and are effective on systems where there is no *mutual* authentication. In most contactless smart card systems, there is usually only unidirectional authentication where the smart card is required to authenticate to the card terminal but the card terminal is not required to authenticate to the smart card. In these systems, the fraudulent merchant can use a fake card terminal to interact with genuine cards. If there is mutual authentication, the smart card will be able to determine that the card terminal is not genuine and terminate the transaction.

### **3. Denial of Service (DOS)**

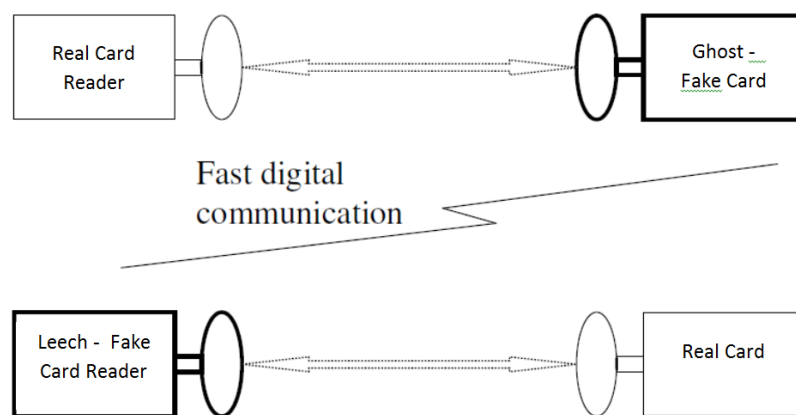
All wireless communications, including contactless smart cards, are susceptible to jamming. Random (noise) signals can be transmitted at the same frequency as the communication signals that inhibit the receiving antennas' ability to interpret the data, resulting in DOS. Incomplete transactions may tie-up the resources of the card terminal, causing the card terminal to be incapable of handling more transactions and resulting in DOS as well.

DOS is an attack on the availability of the system. The only means to prevent jamming is to place the system in a Faraday cage, which is infeasible. However, systems can be designed to minimize the effect of jamming such as

allowing for the expeditious clearing of incomplete transactions upon timeout in order to free the card terminal resources for more transactions.

#### 4. Relay Attacks

Relay attack is used to trick the reader into communicating with a victim smart card that is very far away. The basic relay attack system consists of a ghost (i.e., a fake card) and a leech (i.e., a fake card reader) to create bidirectional communications between the victim's smart card and a real card terminal for transactions. The ghost is used to establish communication between a fake smart card and a real card terminal while the leech is used to establish communication between a real smart card and a fake card terminal. This relay allows bidirectional flow of information between the victim's smart card and the real card terminal to make transactions. The basic system setup for relay attack is shown in Figure 1. A typical relay attack is to charge someone else's credit card for a purchase [17].



**Figure 1. Basic System Setup for Relay Attack [18] (modified)**

Relay attack is an attack on the integrity of the system and has serious security implications as the attacker is able to circumvent most cryptography techniques since the attacker does not need to interpret the information that is exchanged. The attackers only need to relay and replay the information and ensure that the genuine smart cards and the card terminals receive what they are

expecting. The basic countermeasure to prevent a relay attack is to enforce a maximum response time or to use distance-bounding protocols. Yet, while distance-bounding protocols are the most effective countermeasure, these protocols require accurate estimates of the distance between the card and the card terminal, which are hard to implement and hence, these protocols are not commonly used [18].

## **5. Radio Frequency Analysis**

Radio frequency analysis is a side channel attack that is a mixture of power analysis and electromagnetic analysis. This attack measures the variations in the electromagnetic field surrounding the smart card while it is in operation in order to derive the information that the smart card is processing. This is possible because the electromagnetic field surrounding the smart card varies according to the power consumed by the smart card, which is dependent on the type of information or particular operation that the smart card is processing. Hence, by performing reverse engineering, the variations in the electromagnetic field can be used to determine the variations in the power consumption and this data in turn can be used to infer the information that the smart card is processing.

Radio frequency analysis is an attack on the confidentiality of the system and exploits the vulnerability of contactless communication which is that it is easy to intercept. A powerful countermeasure to this attack is to use message and exponent randomization when programming contactless smart cards such that useful information cannot be easily derived just by analyzing the variations in the electromagnetic field [16].

## **F. SECURITY REQUIREMENTS**

As described in previous sections, the enormous benefits of employing contactless smart card systems drive the increasing use of these systems. However, there are also security concerns regarding the use of these systems that need to be addressed. Security controls should be in place to protect the

confidentiality, integrity, and availability (CIA) of smart card systems and to mitigate the security risk associated with the use of these systems. As most of the attacks discussed previously exploit the vulnerabilities in the contactless communication system, this thesis will focus on the security protection of contactless communication (i.e., protecting the bits in transit between the smart cards and the card terminals).

## **1. Confidentiality**

Confidentiality is the assurance that there is no unauthorized disclosure of information and protects against eavesdropping. The common approach to information confidentiality is to use strong cryptography techniques to encrypt the information before sending it, so that the information is not sent as plaintext. Alternatively, another confidentiality tactic is to perform the contactless transactions in a shielded room containing only trusted equipment operated by trusted subjects. This approach is very costly and is only adopted by organizations that need to protect very sensitive and classified contactless transactions.

## **2. Integrity**

Integrity is the assurance that there is no unauthorized modification of information and there is information authenticity. To ensure integrity in contactless smart card systems, the information exchange needs to be checked for improper modifications and there is a need for mutual authentication between the smart cards and the card terminals. The common approach is to use strong cryptographic techniques. For example, a digital fingerprint of the information that is sent can be used to check for improper modifications. This requires the creation of secret keys that are needed to provide reliable indicators of the identities of the devices at both ends of the contactless session. Overall, integrity protects against unauthorized modifications, replay attacks, and covert transactions.

### **3. Availability**

Availability is the assurance that authorized users, processes, and devices have timely and reliable access to information and services. Unlike confidentiality and integrity, cryptographic techniques cannot be used to ensure availability. Availability can only be improved by having a redundant set of systems. Defensive programming can also be used to improve availability by programming the card terminals to purge incomplete transactions upon timeout in order to free resources for more transactions and mitigate attempted DOS attacks.

### **4. Non-repudiation**

Non-repudiation is the assurance that an individual cannot deny having participated in the transactions. Asymmetric cryptography (i.e., public key cryptography, or PKC) is a possible method to achieve electronic non-repudiation. PKC can be used to prove that a smart card is used in a transaction, but PKC cannot prove that it is the rightful owner making the transaction, as there is a possibility that an attacker could have obtained a victim's smart card.

To counter this threat, PKC needs to be coupled with additional mechanisms such as manual verification of the identity of the presenter of the card by having the operator verify the name printed on the card against a form of ID. Additionally, it is possible to employ multi-factor authentication in a manner such that the card carrier needs to prove his/her identity to the card before the card will enable PKC-based authentication.

Another alternative is record the entire transaction so that there is video evidence that the individual is making the transaction. This is not desirable due to the large overhead incurred in maintaining the equipment and management of the data.

THIS PAGE INTENTIONALLY LEFT BLANK

### **III. OPACITY**

#### **A. OVERVIEW**

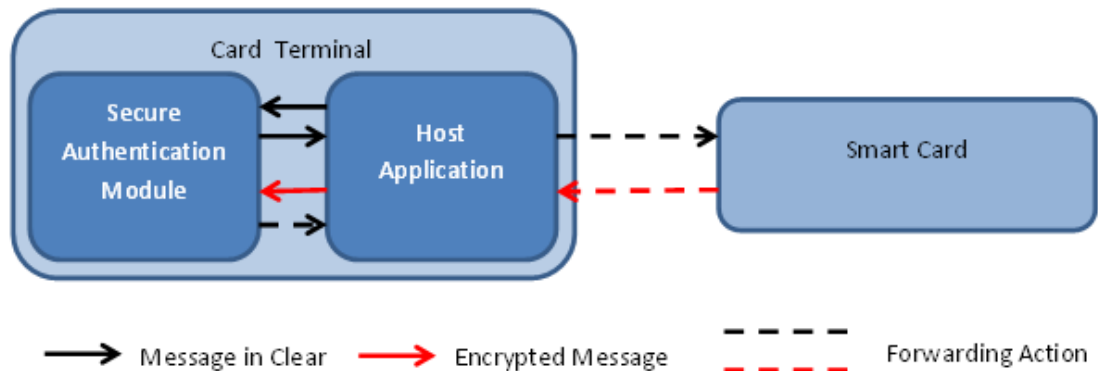
The OPACITY protocol is an open source<sup>1</sup> privacy-enhanced protocol. The protocol is compliant with the guidelines of cryptography and smart card standards and recommendations , which require secure contact or contactless transactions between a smart card and a card terminal (i.e., reader). The protocol is suitable for use in commercial and military authentication applications, such as physical access control to premises, and logical access control to laptops and desktops, as well as ticketing and mass transit applications [19]. The protocol leverages standard cryptographic techniques to perform authentication and to protect the confidentiality and integrity of the data exchanged between the smart card and the card terminal. The protocol has two modes of operation, which are the OPACITY Full Secrecy (OPACITY-FS) mode and the OPACITY Zero Key Management (OPACITY-ZKM) mode. Both of these modes are able to support a range of cipher suites. The operational modes and the cipher suites are configurable to the needs of the operating environment. This thesis will focus on the OPACITY-FS mode optimized for contactless transaction.

A typical setup of the smart card and the card terminal using OPACITY protocol is shown in Figure 2. The card terminal consists of the client application and the Secure Authentication Module (SAM). The client application acts as the interface device to relay data between the smart card and the SAM while the SAM is responsible for all the security functions such as authentication, as well as the encryption and decryption of data.

---

<sup>1</sup> The OPACITY protocol is registered with the U.S. Patent and Trademark Office as a statutory invention to prevent mercenary gains from the commercial sale of this protocol.





**Figure 2. Setup of Smart Card and Card Terminal Using OPACITY Protocol [19] (modified)**

## B. STANDARDS / RECOMMENDATIONS

The OPACITY protocol is compliant with the guidelines of many cryptography and smart card standards and recommendations. Examples include FIPS 140–2 [20], NIST SP 800–56A [21], NIST SP 800–57 Part 1 [22], as well as ISO 24727–4 [23] and ISO 7816–4 [24].

## C. GENERAL CHARACTERISTICS

This section will discuss the general characteristics of the OPACITY protocol. These characteristics include single factor authentication, asymmetric-based authentication and authorization support.

### 1. Single Factor Authentication Protocol

Authentication protocols are designed primarily to perform authentication. Authentication is the process of reliably verifying the claimed identity of a subject (i.e., a human or person) or an entity (i.e., a non-person entity). Mutual authentication ensures that *both* of the subjects or end-entities involved in a particular transaction will authenticate each other prior to the transfer of any sensitive information or execution of any access-controlled operation.

Person authentication and entity authentication are based on different factors of authentication in order to prove the claimed identity. Person authentication is based on proving possession of something known (e.g., passwords), something possessed (e.g., tokens), or something biometrically unique (e.g., fingerprints). A combination of factors can also be employed to achieve multi-factor authentication (e.g., requiring both a token and a PIN). Entity authentication is based on proving possession of something (e.g., shared secrets).

The OPACITY protocol is a single factor authentication protocol that is based on proving possession of something possessed, which is a valid smart card. As all the required authentication information (e.g., identity and cryptographic keys) is stored in the smart card, the claimant will only need to present the smart card to the card terminal for authentication. In addition, as the protocol does not require the claimant to activate the authentication information stored in the smart card, the claimant will obtain successful authentication, even if the claimant is not the rightful owner of the smart card, as long as the claimant proves possession of a valid smart card.

In this case, the smart card is also known as a bearer token. A bearer token is a token in which any claimant in possession of a valid token is able to achieve successful authentication even if the claimant is not the rightful owner of that token. Hence, the authentication protocol may be seen as authenticating the smart card (i.e., entity authentication) rather than authenticating the person in possession of the smart card (i.e., person authentication).

## **2. Asymmetric-Based Authentication**

There are many different types of authentication protocols. The most common type of authentication protocol is the password-based authentication protocol. The password-based authentication protocol is based on something known and makes use of a pre-established shared secret between the subjects or entities involved in the transaction as the basis for authentication. This

protocol is also known as symmetric authentication protocol. The converse of symmetric authentication protocol is asymmetric authentication protocol. An example of asymmetric authentication protocol is a protocol that uses PKC to perform authentication.

The OPACITY protocol is an asymmetric-based authentication protocol that uses PKC [19]. Each smart card and card terminal contains a Card Verifiable Certificate (CVC), a list of root public keys and a static private authentication key. The CVC functions as the public key certificate and is defined according to the X.509 format for public key certificates. The CVC contains the identity of the smart card or the card terminal, the corresponding public key, and other information. The CVC is digitally signed using the Elliptic Curve Digital Signature Algorithm (ECDSA). During authentication, the root public keys are used to verify the digital signature to determine the integrity of the CVC. Before the identity and the public keys on the CVC are used, the corresponding private keys and other information are used to derive symmetric shared secrets and the authentication cryptogram. The authentication cryptogram is the Message Authentication Code (MAC) tag that is used by the card terminal to assess if the authentication process is successful

### **3. Authentication versus Authorization**

Authentication is usually the first step of most sensitive and classified transactions, followed by authorization procedures to verify if the claimants are allowed access to the requested information, systems, or physical infrastructure. A successfully authenticated claimant may not be granted access to the requested resource due to access controls put in place by the resource owner's security policy. Thus, it is important not to confuse authentication with authorization as each accomplishes a different function.

The OPACITY protocol only performs authentication and does not perform any authorization function. However, the authentication information can be used

to support authorization. For instance, the verified identity information can be relied upon to cross-reference against an access control list (ACL) to determine if access should be allowed.

## **D. FEATURES**

The OPACITY protocol is designed with a multitude of features intended to optimize the security and performance of the protocol. This section will provide an overview of these features.

### **1. Security Features**

The protocol provides security features such as mutual authentication and end-to-end protection of the information exchanged between the smart card and the card terminal with forward secrecy and identity privacy [19]. These features help to mitigate the security risk of wireless communications against attacks such as impersonation, identity leak, eavesdropping, and modification attacks.

#### ***a. Authentication***

The protocol is capable of performing mutual authentication in the OPACITY-FS mode to prevent impersonation attacks. In the OPACITY-ZKM mode, the protocol only performs unidirectional authentication. The card terminal authenticates the smart card but the smart card does not authenticate the card terminal.

In general, the OPACITY authentication process involves key agreement, key derivation, and key confirmation steps. The steps are performed using FIPS-approved processes specified in NIST SP800–56A. Key agreement is based on the  $C(1,1)$  steps using the Elliptic Curve Diffie-Hellman (ECDH). OPACITY-FS mode follows a sequence of two  $C(1,1)$  steps while the OPACITY-ZKM mode follows a sequence of one  $C(1,1)$  step. The Key Derivation Function (KDF) and key confirmation steps are the same for the two modes of operation. They are based on a predefined KDF and the  $C(0,2)$  Scheme with Unilateral Key Confirmation provided by the scheme responder to the scheme initiator. The

scheme responder is the smart card while the scheme initiator is the card terminal. The OPACITY-FS authentication transaction diagram is presented in Part 0 of this chapter.

***b. End-to-End Protection with Forward Secrecy***

The protocol provides end-to-end protection of the data exchanged between the smart card and the card terminal with forward secrecy. End-to-end protection protects the confidentiality and integrity of the data and prevents eavesdropping and modification attacks when the transaction is in progress. Protection is achieved using FIPS-approved keyed cryptographic mechanisms such as the Advanced Encryption Standard (AES) cipher to encrypt the data and the Secure Hash Algorithm (SHA) to compute the hashes of the data. The session keys for encryption and hashing are derived using the predefined KDF specified in NIST SP 800–56A.

Forward secrecy enhances security by minimizing the probability that the plaintext can be recovered from captured ciphertext at some later time by the attacker. This protection is accomplished by zeroing all derived session keys once they are no longer used. However, the protocol does not achieve perfect forward secrecy (PFS). This is because in order to achieve PFS, the session keys should not be derivable even if one obtains the long-term static keys that reside on the smart card and the card terminal [25]. As the session keys are derived from the long-term keys for OPACITY protocol, one may recover the session keys if the long-term keys are known using mechanisms such as brute force attack.

***c. Identity Privacy***

The protocol protects the identity of the smart card and/or card owner from identity leak. This prevents the attacker from knowing the smart card or owner identity and associating that identity to any particular transaction.

This protection is achieved by encrypting the smart card's CVC with a symmetric key before transmitting it from the smart card to the card terminal. The derivation of the symmetric key involves using the public key of the card terminal to derive a shared secret using ECDH. Based on ECDH, only a valid card terminal with the corresponding private keys and the same set of ECC domain parameters may derive the same shared secret and derive the symmetric key to decrypt the CVC. Even though the attacker may try to use brute force on the encrypted CVC, this approach is computationally infeasible due to the key's length.

## **2. Performance Features**

The protocol is designed with features such as persistent binding, anti-tearing, and synchronization, as well as simple integration to improve the performance, fault tolerance, and interoperability of the protocol into existing systems [19].

### ***a. Persistent Binding***

The purpose of the persistent binding (PB) feature is to save on the key agreement step for the next transaction. In the current transaction, the protocol computes the shared secrets and one time card identifiers (i.e., the PB records) for deriving the session keys for the next transaction and stores it in the PB table on the smart card and the card terminal, respectively. This is performed for each pair of smart card and card terminal that has completed a successful authentication and has this feature enabled. This feature improves the transaction time for the next transaction.

### ***b. Anti-tearing and Synchronization***

The purpose of the anti-tearing and synchronization feature is to facilitate error recovery. An error can occur when either the smart card or the card terminal fails to receive a response from the other party due to a failed connection or when there is de-synchronization between the PB records that are

stored in the smart card and the card terminal. In the event of a failed connection, the protocol resumes processing from the last successfully completed transaction point (i.e., anti-tearing). In the event of de-synchronization in the PB records, the protocol recovers by repeating the *full* authentication process (i.e., as if it is the first time an authentication is performed for a particular smart card and card terminal). This feature improves the fault tolerance of the protocol.

### ***c. Simple Integration and Interoperability***

The protocol is designed to facilitate simple integration between the client application and the SAM in the card terminal, as well as between the smart card and the card terminal. The SAM is responsible for performing all the security functions (e.g., authentication and encryption of messages) while the client application acts as the interface device in order to relay data between the smart card and the SAM. As such, minimal changes need to be made to the client application in order to use this protocol. In addition, the protocol adheres to the ISO 24727–4 and ISO 7816–4 communication framework that facilitates interoperability between the smart cards and the card terminals from different vendors.

## **E. MODES OF OPERATION**

The OPACITY protocol has two primary modes of operation: OPACITY-FS mode and OPACITY-ZKM mode. The OPACITY-FS mode is optimized to provide secure contactless transactions while the OPACITY-ZKM mode is a lightweight option optimized for both contact and contactless transactions where key management is an issue or where fast transactions are favored over more secure transactions [19].

### **1. OPACITY-FS**

The OPACITY-FS mode is optimized to provide secure contactless transactions. In this mode of operation, the protocol performs mutual authentication. It also provides end-to-end protection, as well as forward secrecy

for sensitive data that are exchanged between the smart card and the card terminal. During the transaction, the identity of the card owner is protected from unauthorized leakage.

## 2. OPACITY-ZKM

The OPACITY-ZKM mode is a lightweight option optimized for both contact and contactless transactions where key management is an issue or where fast transactions are required. In this mode of operation, the card terminal does not need to possess any static key except for the root public key in order to verify the Card Verifiable Certificate Signature. In addition, this mode is faster as it requires only a single C(1,1) step for key agreement as compared to two C(1,1) steps for key agreement for OPACITY-FS. However, there is only unidirectional authentication. The card terminal will authenticate the smart card but the smart card will not authenticate the card terminal. Hence, this mode is only suitable for use in environments where the card terminals are trusted.

## F. CIPHER SUITES

The modes of operations are able to support a range of cipher suites [19] as shown in Table 1.

**Table 1. Cipher Suites Supported By The OPACITY Protocol**

<b>Modes</b>	<b>Fast ZKM only</b>	<b>FS/ZKM</b>	<b>Strong Key Transport</b>	<b>Strong FS</b>	<b>Government Classified</b>
<b><i>Encryption or Mac</i></b>	AES 128	AES 128	AES 256	AES 192	AES 256
<b><i>Smart Card CVC Signature</i></b>	ECDSA 224	ECDSA 256	ECDSA 256	ECDSA 384	ECDSA 384
<b><i>Card Terminal CVC Signature</i></b>	N.A.	ECDSA 256	ECDSA 384	ECDSA 384	ECDSA 384
<b><i>Smart Card</i></b>	ECDH 224	ECDH 256	ECDH 256	ECDH 284	ECDH 384



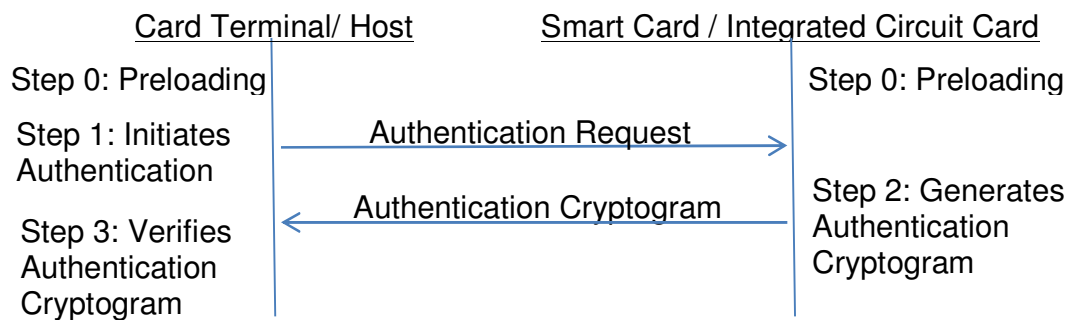
<b>Key Agreement</b>					
<b>Card Terminal Key Agreement</b>	ECDH 224	ECDH 256	ECDH 256	ECDH 284	ECDH 384
<b>Hashing</b>	SHA 1	SHA 256	SHA 384	SHA 384	SHA 384
<b>Nonce</b>	16 bytes	16 bytes	24 bytes	24 bytes	32 bytes

## G. OPACITY-FS AUTHENTICATION

The OPACITY-FS authentication process [19] commences with the card terminal initiating the authentication request. When the smart card receives the authentication request, the smart card will verify the integrity of the authentication information that is received from the card terminal before using the authentication information to derive the authentication cryptogram. The authentication cryptogram is then sent to the card terminal with the smart card authentication information. Likewise, when the card terminal receives the smart card authentication information, the card terminal will verify the integrity of the authentication information before using the authentication information to derive a local copy of the authentication cryptogram. If the two authentication cryptograms match, the authentication is successful. Otherwise, the authentication is unsuccessful.

Prior to the authentication process, the issuer must ensure that the smart card and the card terminal are preloaded with the required authentication information. The authentication process, as shown in Figure 3, consists of four main steps:

- Step 0: Preloading of required authentication information
- Step 1: Card terminal (Host) initiates authentication request
- Step 2: Smart card (ICC) generates authentication cryptogram
- Step 3: Card terminal (Host) verifies authentication cryptogram



**Figure 3. The OPACITY Protocol General Authentication Process**

### **1. Step 0: Preloading of Required Authentication Information**

The smart card (i.e., ICC) and the card terminal (i.e., Host) need to be preloaded with the required authentication information. The authentication information includes the CVC, the private authentication key, other party root public keys for verifying the CVC digital signature, and the ECC domain parameters. The preloaded authentication information is shown in Figure 4.

Remarks	Card Terminal / Host		Smart Card / Integrated Circuit Card (ICC)	Remarks
Card Verifiable Credential (Host Certificate). The CVC contains the following: 1.) Credential Profile Identifier 2.) Issuer identification number 3.) Host identity (ID_sH) 4.) Host public authentication key (Q_sH) 5.) Digital signature (ECDSA-SHA) 6.) Role Identifier	Host CVC (C_H)		ICC CVC (C_ICC)	Card Verifiable Credential (ICC Certificate). The CVC contains the following: 1.) Issuer identification number 2.) ICC identity (ID_slCC) 3.) ICC public authentication key (Q_slCC) 4.) Digital signature (ECDSA-SHA) 5.) Role Identifier
	Host Private authentication Key (d_sH)		ICC Private authentication Key (d_slCC)	
	ICC root public keys (Q_rootlCC)		Host root public keys (Q_rootH)	
Parameters to be generated as specified in ANS X9.62 or from recommendation in FIPS186-3.	ECC domain parameters		ECC domain parameters	Parameters to be generated as specified in ANS X9.62 or from recommendation in FIPS186-3.

**Figure 4. Preloaded Authentication Information**

## **2. Step1: Card Terminal (Host) Initiates Authentication Request**

The Host initiates the authentication process by sending the authentication request to the smart card. Before sending the authentication request, the Host will generate a pair of ephemeral keys, d\_eH and Q\_eH, using ECC key pair generation. d\_eh is the private ephemeral key and Q\_eH is the public ephemeral key. The Host CVC (C\_H) and Q\_eH, are sent as part of the authentication request. The authentication request also includes a control byte (CB\_H) to indicate whether to use the PB feature. This step is shown in Figure 5.

Remarks	Card Terminal / Host		Smart Card / Integrated Circuit Card (ICC)	Remarks
Generate Host ephemeral key pairs using ECC key pair generation.	GEN_KEY_PAIR(d_eH, Q_eH)			
Host Protocol Control Byte (CB_H) - To indicate presence of persistent binding		Authentication Request C_H, Q_eH, CB_H		

**Figure 5. Card Terminal Initiates Authentication Request [19] (modified)**

### 3. Step 2: Smart Card (ICC) Generates Authentication Cryptogram

Upon receiving the authentication request from the Host, the ICC will perform a series of checks and processing, which include validating the Host CVC, verifying the PB feature and key establishments before generating the authentication cryptogram and forwarding it to the Host.

#### *a. Validates Host CVC*

The ICC will verify the digital signature on the C\_H using the Host root public keys (i.e., Q\_rootH) upon receiving the authentication request from the Host. Once the digital signature is verified, the ICC will extract the Host identity (i.e., ID\_sH) and Host public authentication key (Q\_sH) from the C\_H. This step is shown in Figure 6.

Remarks	Card Terminal / Host		Smart Card / Integrated Circuit Card (ICC)	Remarks
			Validate C_H using Q_rootH	Check that certificate is authentic by verifying the digital signature using root public keys
			Extract ID_sH, Q_sH	

**Figure 6. Validate Host CVC [19] (modified)**

#### *b. Verifies Persistent Binding Feature*

Next, the ICC will verify the PB feature by checking the CB\_H. In general, three possible situations will result from this check. The first situation is

that the CB\_H is not set to use the PB feature. The second situation is that the CB\_H is set to use the PB feature but the corresponding persistent record cannot be found in the ICC's PB database. This will occur when it is the first transaction for the ICC-Host pair or when there is an error saving the persistent data in the last transaction. The final situation is that the CB\_H is set to use the PB feature and the corresponding persistent record can be found in the PB database.

In the first and second situations, the ICC will perform the full authentication process. The ICC will verify that the Q\_eH is generated using the same ECC domain parameters before generating its own ephemeral key pairs. It will then perform a sequence of two C(1,1) key agreement steps to derive shared secret Z1 and Z using ECDH as shown in Figure 7. Key K1 is used to encrypt the ICC CVC to protect the ICC's or card owner's identity. The keys, K1 and K2, are generated using a pre-defined KDF.

In the third situation, the ICC will skip the key agreement process and use the shared secret Z and one time ICC identification computed in the previous transaction as shown in Figure 8. In this case, the ICC assumes that the Host also has a valid persistent record and there is no need to send the ICC CVC. Instead, a nonce is generated and sent in place of the ICC CVC. The nonce is also set as key K2.

Remarks	Card Terminal / Host		Smart Card / Integrated Circuit Card (ICC)	Remarks
			If CB_H != PB or CB_H = PB but ID_sH is not registered	<u>Persistent binding check result</u> Case 1: CB_H is not set to use PB + Case 2: ID_sH cannot be found (e.g. first transaction, new transaction or error in saving persistent data in the last transaction).
			Check that Q_eH belongs to same ECC domain	Verify that the Q_eH is generated using the same set of domain ECC parameters as the ICC.
			GEN_KEY_PAIR(d_eICC, Q_eICC)	Generate ICC ephemeral key pair using ECC key pair generation.
			Z1 = EC_DH(d_eICC, Q_sH)	First secret key generated using ECDH. Generated key is used to derive K1 and K2.
			K1 K2 = KDF(Z1, len, info(ID_sH, Q_eICC))	Generate keys K1 (encrypt C_ICC to protect privacy of C_ICC) and K2 (to derive session keys) using pre-approved key derivation function (KDF).
			OpaqueData_ICC = ENCRYPT (AES, K1, C_ICC)	Encrypt C_ICC to protect identity of the smart card owner.
			OTID_ICC = Q_eICC	
			Zeroize Z1, K1.	Erase no longer required keys.
			Z = EC_DH(d_sICC, Q_eH)	Second secret key generated using ECDH. Generated secret key Z is used to derive session keys.

**Figure 7. ICC PB Feature Check Case 1 and Case 2 [19] (modified)**

Remarks	Card Terminal / Host		Smart Card / Integrated Circuit Card (ICC)	Remarks
			If ID_sH is registered	<u>Persistent binding check result</u> Case 3: ID_sH can be found e.g. successful transaction before
			Retrieve Z and OTID_ICC from ID_sH record in the PB registry	
			Generate nonce (N_ICC)	
			OpaqueData_ICC = N_ICC	No need to send C_ICC. Replace with nonce.
			K2 = N_ICC	No need to generate K2. Replace with nonce.

**Figure 8. ICC PB Feature Check Case 3 [19] (modified)**

***c. Derives Session Keys and Generates Authentication Cryptogram***

After completing the PB feature verification process, the ICC will derive the session keys and generate the authentication cryptogram (i.e., AuthCryptogram\_ICC) as shown in Figure 9. The authentication cryptogram is generated using the AES-based MAC algorithm. In addition, the ICC will also store the shared secret (i.e., NextZ) and one-time identification (i.e., NextOTID\_ICC) for the next transaction in the PB database if the PB feature is enabled for the ICC as well as the Host.

The ICC will then send the AuthCryptogram\_ICC, ICC control byte (i.e., CB\_ICC) along with the ICC's authentication information to the Host.

Remarks	Card Terminal / Host	Smart Card / Integrated Circuit Card (ICC)	Remarks
		SK_CFRM  SK_MAC  SK_ENC  SK_RMAC  NextOTID_ICC  NextZ = KDF(Z, len, info(ID_sH, T8(OTID_ICC), T16(Q_eH), K2))	Generate OTID_ICC and Z to be used for next transaction and session keys using pre-approved key derivation function.
		Zeroize Z, K2, d_eICC, Q_eICC	Erase no longer required keys.
		AuthCryptogram_ICC = C-MAC(AES, SK_CFRM, "KC_V_1"  T8(OTID_ICC)  D_sH  T16(Q_eH))	Generate the authentication cryptogram using NIST 800-38B AES based MAC algorithm. Authentication cookie "KC_V_1" indicates ICC is the scheme responder and HOST is the scheme initiator, key confirmation provider and recipient.
		Zeroize SK_CFRM	Erase no longer required keys.
		if CB_H != NO_PB and ICC supports PB and (ID_sH is not registered or CB_H == PB_INIT)	Check if persistent binding (PB) is available and set the control byte (CB) and store the information for the next transaction accordingly.
		Register Z = NextZ for ID_sH	
		if(CB_ICC != PB), CB_ICC = PB_INIT	
		Else CB_ICC = NO_PB	
		OpaqueData_ICC  AuthCryptogram_ICC  CB_ICC  OTID_ICC	
		←	

**Figure 9. Derive Session Keys and Authentication Cryptogram [19] (modified)**



### **3. Step 3: Card Terminal (Host) Verifies Authentication Cryptogram**

Upon receiving the authentication response from the ICC, the Host will perform a series of checks and processing, which include verifying the PB feature and key establishments before generating a local copy of the authentication cryptogram and verifying if the local copy of the authentication cryptogram matches the authentication cryptogram sent by the ICC.

#### ***a. Verifies Persistent Binding Feature***

When the Host receives the respond to the authentication request, the Host will proceed to verify the PB feature. Similar to the ICC PB feature verification outcomes, there are three cases that may result from this action. The first case is that the CB\_ICC is not set to use the PB feature. The second case is that CB\_ICC is set to use the PB feature but the persistent record cannot be found in the Host's PB database. The third case is that the CB\_ICC is set to use the PB feature and the corresponding persistent record can be found.

In the first case, the Host will perform the full authentication process as shown in Figure 10. The Host will verify that Q\_eICC is generated using the same ECC domain parameters. Once verified, the Host will derive the first shared secret, Z1, to derive secret key K1, which is then used to decrypt the encrypted c. Next, the Host will verify the digital signature of the C\_ICC using the list of ICC root public keys (i.e., Q\_rootICC) and from there it will derive the second shared secret, Z.

In the second case, the Host will send a request to the ICC to restart the authentication process and perform the full authentication process as shown in Figure 11. This will occur when there is a de-synchronization between the PB databases and the Host does not have the Z and OTID\_ICC that was computed in the previous transaction to continue with the authentication process.

In the third case, the Host will use the Z and OTID\_ICC in the persistent record to proceed with the authentication process as shown in Figure 12.

Remarks	HOST		Integrated Circuit Card (ICC)	Remarks
<u>Persistent binding check result</u> Case 1: CB_ICC is not set to use PB and OTID_ICC cannot be found.	If OTID_ICC cannot be found (ID_slICC is not registered) and CB_ICC != PB Q_elICC = OTID_ICC			
Verify that the Q_elICC is generated using the same set of domain ECC parameters as Host.	Verify that Q_elICC belongs to the same ECC domain.			
First secret key generated using ECDH. Generated key is used to derive K1 and K2.	$Z1 = EC\_DH(d\_sH, Q\_elICC)$			
Generate keys K1 (decrypt the encrypted C_ICC) and K2 (to derive session keys) using pre-approved key derivation function (KDF).	$K1 K2 = KDF(Z1, len, info(ID\_sH, Q\_elICC))$			
	$C\_ICC = DECRYPT(AES, K1, OpaqueData\_ICC)$			
Check that certificate is authentic by verifying the digital signature using root public keys	Validate C_ICC using Q_rootICC			
	Extract Q_slICC and ICC_cred from C_ICC			
Second secret key generated using ECDH. Generated secret key Z is used to derive session keys.	$Z = EC\_DH(d\_eH, Q\_slICC)$			
Erase no longer required keys.	Zeroize Z1, K1			

**Figure 10. Host PB Feature Check Case 1 [19] (modified)**

Remarks	Card Terminal / Host		Smart Card / Integrated Circuit Card (ICC)	Remarks
Persistent binding check result Case 2: If OTID_ICC cannot be found and yet it can be found on the ICC, there is de-synchronization between the PB registries.	If OTID_ICC cannot be found (ID_sICC is not registered) and CB_ICC = PB			
Erase no longer required keys.	Zeroize d_eH			
Request to restart and perform the full authentication process.	return CB_H = PB_INIT			

**Figure 11. Host PB Feature Check Case 2 [19] (modified)**

Remarks	HOST		Integrated Circuit Card (ICC)	Remarks
Persistent binding check result Case 3: PB record can be found	If OTID_ICC is found (ID_sICC is registered)			
No need to generate K2. Replace with nonce generated by ICC.	K2 = OpaqueData_ICC			
	Retrieve Z and ICC_Cred from OTID_ICC in PB registry			

**Figure 12. Host PB Feature Check Case 3 [19] (modified)**

***b. Derives Session Keys and Verifies Authentication Cryptogram***

After completing the PB verification process, the Host will derive the session keys and generate a local copy of the authentication cryptogram (i.e., AuthCryptogram\_H) using the same input information for the AES-based MAC algorithm as the ICC, as shown in Figure 13. If AuthCryptogram\_H is the same as AuthCryptogram\_ICC, authentication is successful. Otherwise, authentication is not successful and an error message is returned. The authentication cryptogram verification process follows the C(0,2) Scheme with Unilateral Key Confirmation provided by scheme responder to scheme initiator.

After a successful authentication, the Host will store the shared secret (i.e., NextZ) and one-time identification (i.e., NextOTID\_ICC) for the next transaction in the PB database as long as the PB feature is enabled for the Host as well as the ICC.

Remarks	HOST		Integrated Circuit Card (ICC)	Remarks
Generate OTID_ICC and Z to be used for next transaction and session keys using pre-approved key derivation function.	SK_CFRM  SK_MAC  SK_ENC  SK_RMAC  NextOTID_ICC  NextZ = KDF(Z, len, info(ID_sH, T8(OTID_ICC), T16(Q_eH), K2))			
	Zeroize Z, K2, d_eH, Q_eH			
Generate the authentication cryptogram using NIST 800-38B AES based MAC algorithm. Authentication cookie "KC_V_1" indicates ICC is the scheme responder and HOST is the scheme initiator, key confirmation provider and recipient.	AuthCryptogram_H = C-MAC(AES, SK_CFRM, "KC_V_1"  T8(OTID_ICC)  ID_sH  T16(Q_eH))			
	Check if the AuthCryptogram_ICC = AuthCryptogram_H			
	If check fails, return Auth_Error			
	Zeroize SK_CFRM			
Check if persistent binding (PB) is available and set the control byte (CB) and store the information for the next transaction accordingly.	If CB_ICC != PB_INIT			
	Register ICC_Cred and Z = NextZ for OTID_ICC = NextOTID_ICC			
	Compute PB address			
	CB_H = PB			
	Else PB address == NULL			
	CB_H = NO_PB			

**Figure 13. Host Verifies Authentication Cryptogram [19] (modified)**

## H. STRENGTHS AND LIMITATIONS

This section presents an analysis of the strengths and limitations of the OPACITY protocol. This analysis will be conducted at the protocol level and will not address any details of the cryptographic algorithms that are used in the protocol, such as SHA and AES. As a starting point, it is assumed that these heavily vetted and NSA-approved algorithms function securely “as advertised.”

### 1. Strengths

#### *a. Asymmetric Cryptography*

The OPACITY protocol is an asymmetric-based authentication protocol that uses PKC. PKC provides numerous advantages:

(1) Minimize key distribution. One of the key advantages of PKC is that it minimizes the key distribution problem that is so prevalent with symmetric cryptography. In a symmetric key system of  $n$  users, if all the  $n$  users are required to communicate securely with one another, each user must have  $(n - 1)$  secret keys. This amounts to a total of  $n(n - 1)/2$  keys that need to be generated and distributed securely among the  $n$  users. With asymmetric cryptography, the key distribution “problem” is reduced to how each of the  $n$  users may obtain his/her private key securely. This is often done in-person at an office approved for this purpose. Once each of the users has obtained his/her respective private key and corresponding certified public key, there is no key distribution problem per se, as the public keys can be stored and shared/distributed without any protection.

(2) No need for pre-defined shared secret. Another advantage of PKC is that there is no need to predefine any shared secrets for authentication [26]. Key agreement algorithms such as ECDH can be used to establish shared secrets to derive session keys on the fly during the authentication process. This also helps to minimize the distribution problem described in the previous sub-section.

(3) Strong identification. A third advantage of PKC is its ability to use a digital signature. A digital signature provides two important functions [26]. Firstly, a digital signature proves who generated the information. Secondly, a digital signature protects against unnoticed information modification.

For OPACITY protocol, the CVC is signed by the issuer and provides strong assurance that the identification information and public authentication key are authentic and have not been modified.

#### ***b. Use of Different Session Keys***

In the OPACITY-FS mode, different session keys are used for encryption, hashing (i.e., MAC), and to generate the authentication cryptogram. The use of different keys for encryption and hashing helps to mitigate plaintext—ciphertext pair attacks. In addition, the use of different keys also helps to mitigate modification attacks as the attacker needs to crack both keys in order to perform a successful attack, which is computationally impossible given the short transactional time.

#### ***c. Mutual Authentication and End-end Protection with Forward Secrecy and Identity Privacy***

The OPACITY protocol performs mutual authentication and provides end-end protection, mitigating a number of vulnerabilities in contactless transactions. For instance, the protocol is resistant against eavesdropping, modification, impersonation, and most man-in-the-middle attacks. In addition, the protocol also provides identity privacy by encrypting the smart card's ICC before transmitting.

## **2. Limitations**

#### ***a. Asymmetric Authenticating***

Despite the numerous advantages of asymmetric cryptography and PKC, there are also limitations.

(1) Higher latency. A key limitation of asymmetric cryptography is that it has a slower performance than symmetric cryptography. For this reason, there is research interest in developing hybrid authentication protocols that use a mixture of asymmetric and symmetric techniques.

(2) No CVC revocation. Another limitation of OPACITY protocol is that there is no CVC revocation functionality in the current design [19]. The purpose of the CVC revocation functionality is to mark the CVC as “invalid” or “revoked” so that the public keys (i.e., the user’s public keys contained in the CVC and the card’s public authentication key), as well as the corresponding private keys, can no longer be used in the event of loss or theft of a smart card. The issuer will only need to regenerate, issue, and update the backend system with the new public keys, while minimizing changes to the backend system. Without the CVC revocation functionality, the user identification information and associated authorization privileges, in addition to the keys, need to be removed from the backend system, regenerated, issued, and updated to the backend system, incurring substantial changes to the backend system.

#### ***b. Single Factor Authentication***

This protocol is a single factor (i.e., the subject *has* the card) authentication protocol and it is not able to achieve the security confidence of the multi-factor authentication protocol.

Based on the guidelines provided in NIST SP 800–116 [27] for physical access control system (PACS), this protocol provides some confidence for PACS and is only suitable for physical access control to controlled areas. In order to be suitable for use to limited or exclusive areas, the protocol has to be combined with other factors (i.e., something you *know* or something you *are*) of authentication to achieve high or very high confidence.

#### ***c. Bearer token***

(1) Does not achieve user on-repudiation. The application of this protocol alone without other security mechanisms makes the

smart card a bearer token. As such, anyone with mere possession of a valid smart card is able to be successfully authenticated. Hence, the protocol does not achieve user non-repudiation even though the protocol uses PKC. The protocol only achieves non-repudiation indicating that a particular smart card is used. To achieve user non-repudiation the protocol needs to be combined with other factors of authentication, such as explicit user activation of the smart card through biometric means, before the smart card can be used for authentication. Since biometrically unique features uniquely identify each individual, the individual cannot deny making the transaction when the smart card is used for authentication, therefore achieving user non-repudiation.

(2) No proof of control of token. Based on the guidelines provided in NIST SP 800–63–1 [28] for electronic authentication, one of the criteria to achieve a minimum assurance level for electronic authentication is that the claimant must be able to prove possession and control of the token that is used in the authentication process. Based on this guideline, the protocol does not achieve the minimum assurance level as the protocol does not require the claimant to prove control of the smart card



THIS PAGE INTENTIONALLY LEFT BLANK

## **IV. PLAID**

### **A. OVERVIEW**

Developed by Centrelink, an Australian Government Statutory Agency, the PLAID protocol is a common, non-proprietary smartcard authentication protocol that is suitable for both PACS and logical access control (LACS). This protocol is designed to bridge the gap between existing RFID-based technologies that offer speed but lack the necessary security features, and PKI-based authentication, which is cryptographically secure but lacks the speed necessary in many contactless smartcard scenarios. The PLAID protocol utilizes a hybrid standards-based symmetric and asymmetric cryptography to protect data transmissions between the smartcard and terminal devices. In addition, this protocol is designed to perform high strength mutual authentication in less than 0.3 of a second, thus preventing the leakage of any form of information that might prove useful for an attacker.

According to the developers, stringent evaluations have been conducted by renowned cryptographic organizations in the hope that by improving consumer confidence the number of commercial end-users and vendors that implement the PLAID protocol within their products will improve. The PLAID protocol is an extremely versatile protocol and can be customized to support either single or dual factor authentication.

### **B. STANDARDS/RECOMMENDATIONS**

As the PLAID protocol is relatively new, it is not formally recognized as a protocol that adheres to international standards such as ISO 24727–4 or recommendations made in FIPS 140–2. However, with the latest version of the PLAID Specification (version 8), enhancements and simplifications have been made to support ISO 24727 parts 3 and 6. The current version of the protocol has been formally mapped to the Australian Standard (AS 5185–2010) which will subsequently be encompassed as part of the ISO/IEC standard [5]. As the PLAID

protocol has the necessary components required to provide strong authentication and security for both PACS and LACS implementation, it is still able to meet the criteria set by both the ISO/IEC 14443 and ISO/IEC 7816 standards.

## **C. GENERAL CHARACTERISTICS**

This section will discuss the general characteristics of the PLAID protocol.

### **1. Multi-Factor Authentication**

The PLAID protocol is extremely flexible in its implementation and can support both single-factor or multi-factor authentication. The primary factor of authentication is proof of possession of something the user *has* (i.e., a valid smart card). Multi-factor authentication is achieved by incorporating two additional factors of authentication, which is something the user *knows* (e.g., a PIN) and something the user *is* (e.g., fingerprint), in addition to proof of possession of a valid smart card. Multi-factor authentication provides higher confidence in the asserted identity of the claimant.

### **2. Symmetric and Asymmetric Key Algorithms**

Symmetric key algorithms are a class of algorithms that utilize the same cryptographic keys for both encryption and decryption of data. These keys represent a shared secret between the parties involved in the exchange of information. One drawback however, is that both parties have access to the secret key, which not only increases the likelihood of compromise, but also precludes its usage in applications that require the security objective of non-repudiation.

Asymmetric key algorithms on the other hand, refer to a class of algorithms that require a pair of separate keys, of which one is a secret and the other is public. Despite the keys being different, the two keys are actually mathematically related. What one key encrypts the other can decrypt. Neither of these keys can be used to perform both functions for any particular object, and therefore cannot be used in isolation.

For the purposes of authentication, the PLAID protocol utilizes both symmetric and asymmetric key algorithms. To ensure the authenticity of the receiving party, an asymmetric algorithm, in the form of RSA, is first used to encrypt certain selected identification data. Once the initial authentication has been established, the PLAID protocol uses symmetric encryption for all subsequent data transactions as it offers improvements in performance. Specific details of how the protocol works and the rationale for having both asymmetric and symmetric algorithms in the protocol will be described in the subsequent section.

### **3. Authentication versus Authorization**

As mentioned in the previous chapter, authentication is the mechanism whereby a system is able to reliably identify users, while authorization is the mechanism by which the system determines what level of access a particular authenticated user is permitted (as specified in the access control portion of a given security policy).

In the case of the PLAID protocol, the protocol offers both forms of functionalities. Authentication is accomplished through various means, such as ensuring the IFD is able to correctly decrypt the RSA encrypted data from the ICC, as well as having the IFD perform stringent checks to ensure that the ICC holder has entered the correct PIN or biometric profile. Depending on the implementation of the system, authorization is provided by checking the operational mode identifier (OpModeID) stored in the ICC against the access control system (ACS) record. For example, an implementation might utilize the OpModeID to grant a user physical access to rooms that are of a less sensitive nature while restricting them from rooms with a higher security classification, despite them being valid—authenticated—users.

## **D. FEATURES**

The PLAID protocol is designed with multiple features that optimize the security and performance of the protocol. This section will provide an overview of these features.

### **1. Security Features**

The protocol provides security features such as mutual authentication and end-to-end protection of information between smart card and card terminal with identity privacy. Such features are useful to mitigate against the security risks commonly found in smart cards, such as man-in the middle (MITM) attacks, replay attacks and data leakages.

#### ***a. Mutual Authentication***

The protocol is capable of performing mutual authentication to prevent impersonation attacks from either card or reader. In general, the authentication process involves key agreement, key derivation and key confirmation steps, which are in accordance to guidelines stipulated in the NIST SP800–56A and ISO/IEC 9798–2 document.

Here, key agreement is based on the negotiation of the key sets, where the IFD, in the initial authentication command, gives the ICC a list of supported key sets in preferential order. The ICC will then utilize the most preferred set based on the order of priority. Next, key derivation is achieved when the IFD uses the diversification data to determine the final authentication key. This will be explained in greater clarity in the subsequent section.

#### ***b. End-to-End Protection***

The PLAID protocol is able to provide end-to-end protection of the exchanged data as it utilizes a suite of FIPS-approved keyed cryptographic mechanisms such as RSA and AES cryptography. As an added measure, SHA is also used to hash data in order to derive the final authentication key. These

mechanisms prevent the possibility of malicious attacks as no identifiable or unique information is transmitted in the clear. Furthermore, the use of SHA acts as a countermeasure that prevents any “man-in-the-middle” device from modifying session data without detection.

***c. Authorization Checks***

The PLAID protocol achieves authorization via the OpModelID, which specifies the cardholder’s privileges.

**2. Performance Features**

The protocol is designed to be lightweight while still being cryptographically stronger and faster than existing authentication protocols. It offers performance features such as easy integration and interoperability.

***a. Simple Integration and Interoperability***

The protocol is designed using existing off the shelf symmetric and asymmetric algorithms, which allows for easy integration with existing systems. Furthermore, the fact that it is lightweight means that it can be efficiently implemented on current generation hardware. Previously, the use of asymmetric algorithms on smart cards was considered too slow to use for physical access control. With the PLAID protocol, authentication can be completed in as little as 200ms. These results are not theoretical, as tests have been conducted on production cards to ensure that such speeds are achieved in actual practice.

**E. MODES OF OPERATIONS**

Using the two-byte ACS record that is sent to ICC in the final authentication command as a reference, the PLAID protocol allows for up to 65535 operational modes. This functions similar to that of a capability list where a different and distinct ACS record can be verified by either (or both) the IFD or a backend database depending on how the system is implemented. For example, an individual might use the ICC for both physical and logical access. Depending

on the record required by the reader, the protocol is able to provide an authenticated record of the type required for the particular environment. The records could range from a Weigand number, an ICAO credential, a RFC 4122 UUID record or even a biometric template depending on implementation. Having these different modes of operation supports both physical and logical access control in many environments.

#### **F. SUGGESTED KEY LENGTHS AND ALGORITHMS**

As there are often tradeoffs between key length, cryptographic options and speed, the versatility of the PLAID protocol allows it to be implemented with various key lengths, cryptographic algorithms, and modes, depending on the various performance requirements of the user. Table 2 presents some of the suggested key lengths and cryptographic algorithms:

**Table 2. Cipher Suites Supported By The PLAID Protocol**

<b>Symmetric Algorithm</b>	<b>Key Length (Bits)</b>	<b>Mode</b>	<b>Asymmetric Algorithm</b>	<b>Key Length (Bits)</b>	<b>Target Transaction Time (ms)</b>
AES	128	CBC	RSA	1024	200
AES	192	CBC	RSA	1536	350
AES	256	CBC	RSA	1984	480
AES	256	CBC	RSA	2048	500

#### **G. AUTHENTICATION PROCESS**

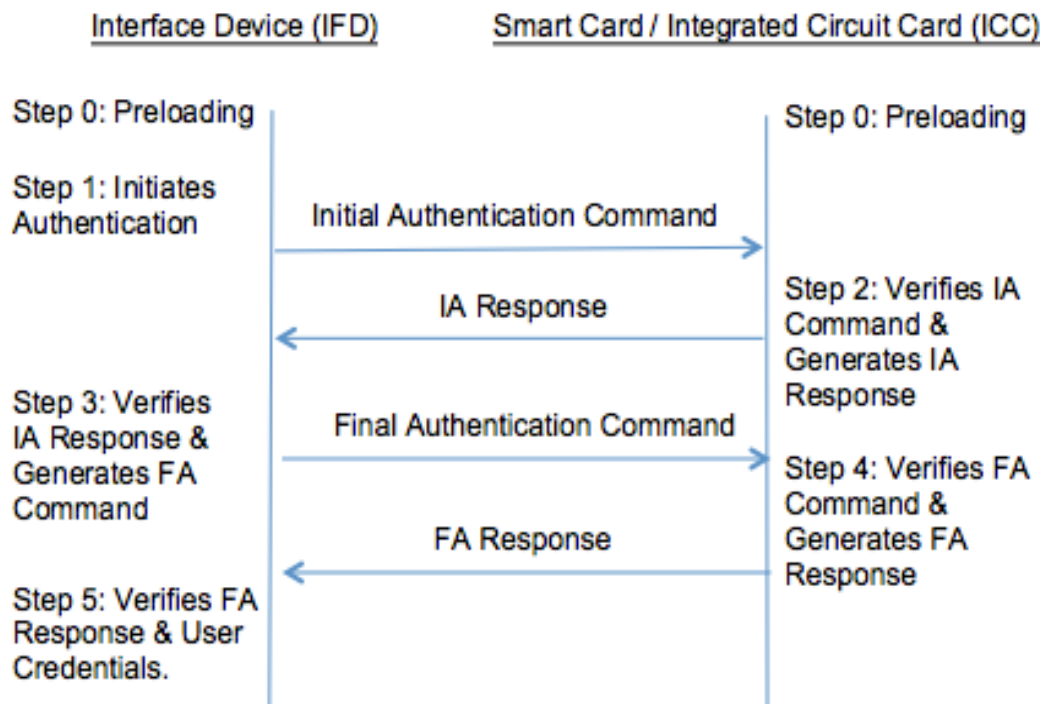
The authentication process is considered relatively straightforward. Developers of the protocol designed the process to be simple as the rationale was to optimize performance and reduce complexity. When the ICC receives an initial authentication request command from the IFD, the ICC will verify the authenticity of the authentication information it received from the IFD. It does this by generating a response using randomly generated numbers, embedded with

data required by the initial authentication command and an asymmetric key. This response is sent back to the IFD where the IFD then has to decrypt the response using the initial authentication key. Upon successful decryption, the IFD creates yet another authentication command using random numbers and the symmetric key. Successful authentication is only achieved when the ICC is able to successfully decrypt the final authentication command. Again, depending on how this system is implemented, the final step consists of the ICC sending information, such as a PIN or biometric information, in its response to the IFD to provide a greater level of confidence in the identity of the cardholder. This confidence can then be leveraged to support access control via reference to user privileges carried on the card.

The described authentication process, as shown in Figure 14, consists of six main steps, which will be described in detail in the subsequent sections:

- Step 0: Preloading of required authentication information
- Step 1: IFD initiates initial authentication (IA) command
- Step 2: ICC generates authentication response to the IA command
- Step 3: IFD verifies IA response and generates final authentication (FA) command to ICC using symmetric keys
- Step 4: ICC decrypts FA command and sends back an encrypted FA response, containing personal information such as PINHash/biometric data to the IFD for authorization purposes
- Step 5: Depending on implementation, the IFD decrypts the FA response and performs some backend checks to verify the information before granting access to the user





**Figure 14. The PLAID Protocol General Authentication Process**

#### **1. Step 0: Preloading of Required Authentication Information**

For the PLAID protocol to function, both the IFD and ICC are required to be preloaded with the relevant authentication information. The authentication information includes a list of key set identifier (KeySetID). The KeySetID is used to identify the set of keys that will be used for authentication. In addition, the ICC is required to be preloaded with key diversification data (DivData) and an ACS record. The DivData is essentially an 8-byte number that is set at each instantiation for use in the key diversification algorithm. This is necessary due to the fact that symmetric keys are involved. Having the DivData prevents a breach of the system master keys in the event an ICC is lost. Next, the ACS Record is used together with the OpModelID for the purpose of authorization by the backend PACS or LACS access control system. This prevents legitimate users from obtaining access to places where they are not authorized due to insufficient privileges.

## **2. Step 1: IFD Initiates Authentication Command**

The authentication process begins with the IFD initiating the first authentication request. It does this by sending the ICC an initial authentication command in the form of an application data protocol unit (APDU) requesting it for the DivData that is pre-stored in the ICC. In the context of smart cards, an APDU is the communication unit between the IFD and the ICC. There are generally two categories of APDUs, the command and the response. The body of the APDU contains the list of authorized KeySetIDs that has been encoded using an Abstract Syntax Notation Number (ASN.1) and is hierarchically ordered with the preferred KeySetID appearing first. For the purposes of reducing complexity, the theory of how ASN.1 works, will not be explained in this thesis. Instead, the point that is significant is how ASN.1 is initially used to synchronize both devices prior to the use of encryption. The primary purpose of the authentication command is to protect the privacy of the DivData that is required to diversify the keys in the subsequent authentication command in order to prevent them from being sent in the clear.

## **3. Step 2: ICC Responds to the IA Command by Generating IA Response**

Upon receiving the initial authentication command, the ICC parses the list of KeySetIDs and retrieves the first initial authentication key (IAKey) that matches a KeySetID supported by the ICC. A unique security feature that the PLAID protocol offers is the use of a ShillKey, which prevents any indication that an error has occurred in the event none of the KeySetIDs match a key stored by the ICC. The ICC still sends a random byte encrypted ShillKey back to the IFD indicating that a transaction has occurred. This minimizes the amount of useful information an attacker might be able to obtain from generated error codes in the event of failed transactions.

According to the specifications [5] produced by the developers of the protocol, a true random number generator (TRNG) is used to produce a random

number (RND1). However, according to the Australian Standards AS 5185–2010 document, the word “true” is dropped and only the term random number generator is used. This thesis agrees with the standards and argues that a random number generator based solely on deterministic computation cannot be regarded as “true” since its output is inherently predictable. That being said, the problem of distinguishing the difference between output of a “true” random number generator and a pseudo-random number generator is indeed very difficult and will not be discussed in this thesis. The assumption here however, is that the random number generator (RNG) is a piece of hardware embedded in the card that generates random numbers from a physical process such as thermal noise or other quantum phenomena to produce an output. The size of this random number is based on the key size of the selected AES cipher. In this case, if the selected AES cipher is 32 bytes, then the size of RND1 would be identical.

Upon the generation of RND1, the ICC proceeds to retrieve the DivData to produce a string (STR1). Therefore, the contents of STR1 contain the KeySetID, DivData and two copies of RND1. The purpose of having a repeat of RND1 in STR1 is for it to function as a checksum. The next step the ICC performs is to encrypt STR1 using the RSA algorithm to output the encrypted string,  $ESTR1 = \text{RSA}_{\text{ENCRYPT}}^{\text{IAKEY}}(\text{STR1})$ . The IAKey functions as the asymmetric key because the encryption process utilizes only its modulus and public exponent. ESTR1 is then transmitted, in the form of an IA response back to the IFD. The use of the asymmetric cipher prevents an adversary from decrypting the message over the air in the event a card is compromised.

#### 4. Step 3: IFD Responds to the IA Response from ICC by Generating FA Command

Upon receiving the IA Response from the ICC, the IFD proceeds to decrypt it using the following steps. Firstly, the IFD calculates ESTR1 by using the first KeySetID identified in the ASN.1 list. This can be illustrated in the manner where  $\text{STR1} = \text{RSA}_{\text{DECRYPT}}^{\text{IAKEY}}(\text{ESTR1})$ . Next, the IFD compares both

copies of the generated RND1 that was derived from decrypting ESTR1 to ensure that the decryption was successfully performed. In the event that the process was unsuccessful, the IFD utilizes the next set of KeySetID in the ASN.1 list. This process is repeated until decryption is achieved or all available KeySetIDs have been attempted. As the same asymmetric keys might be utilized in multiple KeySetIDs for large implementations, there is a need for the IFD to extract both the KeySetID and the DivData from STR1 to determine which key set is being used. In addition, if the IFD has successfully decrypted and validated the encrypted string, then the ICC would have proven knowledge about an approved IKey value.

Upon obtaining all the required parameters, the IFD then proceeds to generate a new random number, RND2, using the RNG. Similar to the random number, RND1, produced by the ICC, the size of RND2 is identical to the key size of the selected AES cipher. However, RND2 is not used on its own. Instead, the IFD calculates a new composite number, RND3, by combining the hash of both RND2 and RND1, where  $RND3 = \text{SHA}(RND1 \parallel RND2)$ . Generating RND3 prevents the possibility of man-in-the-middle attacks and guarantees that a malicious device would not be able to intercept the communication between both the initial authentication and final authentication steps. Next, the IFD utilizes the DivData to derive the final authentication key, FAKey, where  $\text{FAKey} = \text{AES}_{\text{ENCRYPT}}^{\text{FAKey}}(\text{DivData})$ . Here, it is important to notice that the FAKey differs from the IKey as it uses the symmetric AES cipher to achieve the required performance with a longer key length than current asymmetric ciphers are able to achieve. Hence, it can be said that the PLAID protocol uses a hybrid cryptography method.

Lastly, the IFD creates a new authentication message, STR2, where  $\text{STR2} = \text{OpsModelID} \parallel \text{RND2} \parallel \text{RND3}$ . The OpsModelID is sent in the new authentication message as it aids the ICC in determining the type of system the IFD is and helps the ICC load the correct records or payload. The message is

then encrypted with the FAKey to produce the FA Command, where  $ESTR2 = AES_{ENCRYPT}^{FAKEY}(STR2)$ , which is then sent over the air back to the ICC. This concludes step 3 of the process.

#### **5. Step 4: ICC Response to the FA Command**

Upon receiving the FA Command, the ICC proceeds to derive the authentication message through calculation where  $STR2 = AES_{DECRYPT}^{FAKEY}(ESTR2)$ . It is important to know that the FAKey that is to be used, is determined using the KeySetID that was previously agreed upon in Step 2. As the ICC does not know what RND3 is, there is a need to derive the number using the hash of both RND1, which it already knows in Step 2, and RND2, which it obtained from the decrypted STR2. Upon determining RND3, the ICC checks its RND3 with the RND3 obtained from STR2. If the ICC successfully decrypts the message and validates RND3, then the ICC will consider the IFD authenticated. However, in the case that RND3 is not validated, the PLAID protocol has a special feature where the ICC responds using a random byte encrypted with a ShillKey. This prevents the possibility of leaking error information to attackers.

Next, the ICC retrieves the appropriate fields such as the ACSRecords, based on the OpModelID extracted from STR2. The ICC then proceeds to create a new bit string, STR3, where  $STR3 = DivData \mid ACSRecord \mid (Null, PINHash \text{ and/or } Biometric \text{ Minutiae})$ . The reason why STR3 may contain a PINHash, a biometric minutiae or even an empty string is because it is highly dependent on how a system is being implemented. For environments that require tighter access control as opposed to a place where there is no need to verify user credentials, a combination of both a PINHash and minutiae could be required. STR3 is then encrypted to produce a FA Response where  $ESTR3 = AES_{ENCRYPT}^{RND3}(STR3)$ . The cipher mode for this operation as well as the previous cipher mode in Step 3 must be cipher block chaining (CBC) mode. The FA Response is then transmitted back to the IFD.

## **6. Step 5: IFD Processes Credentials**

The IFD proceeds to decrypt the FA Response by calculating STR3 where  $STR3 = AES_{DECRYPT}^{RND3}(ESTR3)$ . Next, the IFD compares the transmitted DivData sent in STR3 with the DivData that was sent in the initial authentication command. In the event both sets of DivData do not match, then the authentication process fails. Depending on implementation, if PIN authentication is required, then the IFD will have the cardholder's PINHash in STR3 and it will compare the PINhash to the hash of the PIN input by the cardholder at the IFD. Similarly, if biometric authentication is required, the biometric information presented by the cardholder to the IFD is compared with the minutiae retrieved from STR3. Throughout this process, if any of the cardholder's information does not match the information found in STR3, the authentication fails.

Lastly, the ACSRecord is extracted from STR3 and mutual authentication is achieved. This record is then sent to the implemented back office system that is appropriate for controlling whatever access is being requested (e.g., opening a door). This ensures that a cardholder has sufficient privileges to access a particular object (physical or logical) once his/her identity has been verified.

## **H. STRENGTHS AND LIMITATIONS**

This section presents an analysis of the strengths and limitations of the PLAID protocol. Similar to the analysis conducted in the earlier chapter, this analysis will be conducted at the protocol level, and hence will not address any details of the cryptographic algorithms that have been utilized in the protocol. The assumptions made here are again similar to those made previously that all algorithms function as specified and are secure.

## **1. Strengths**

### ***a. Hybrid Cryptography***

The PLAID protocol utilizes both asymmetric and symmetric based cryptography. Having a hybrid of both kinds of cryptography provides a wide variety of advantages.

Firstly, the advantages of using asymmetric-based cryptography to establish the initial authentication are examined. Using asymmetric keys for initial authentication minimizes the key distribution problem. The next advantage of the PLAID protocol using asymmetric encryption is that both the IFD and ICC do not need to have their secret shared between each other in order to communicate. Hence, in the event a particular card is compromised, an adversary would still be unable to decrypt the over-the-air traffic from any other cards. Although this form of encryption is strong and effective, asymmetric algorithms are comparatively more complex and computationally more expensive. These mean that the messages will take a longer time to encrypt and decrypt.

Therefore, the PLAID protocol uses symmetric encryption for the second part of the process. Symmetric keys perform much faster when compared to asymmetric keys. In addition, this form of encryption is easy to use and simple to carry out. The protocol only needs to specify and share the secret key begin encrypting and decrypting messages. Many implementations utilize asymmetric encryption to encode a symmetric key and then transfer it to the other party. The symmetric key is then used to transmit the actual message which is much more efficient in CPU utilization. It would seem that the PLAID protocol has adopted a similar methodology, as the initial authentication command uses asymmetric encryption to encode the DivData and random number, which are used to produce the symmetric key that will be subsequently used.

**b. Multi-Factor Authentication**

The PLAID protocol allows the use of multiple-factor authentication as the protocol not only requires the user *have* the card, but can also require the submission of a PIN (*know*), or a biometric (*is*). Having multi-factor authentication significantly decreases the probability that the requestor is presenting false evidence of its identity.

**c. Mutual Authentication, End-To-End Protection**

The PLAID protocol performs mutual authentication and provides end-to-end protection by providing a suite of mechanisms against vulnerabilities in current contactless transactions. For instance, having the protocol respond with a Shillkey instead of an error message prevents information from being leaked to an attacker performing a brute force attack, by removing any indication that an error has occurred. Next, the key diversification algorithm ensures that a system remains secure even in the event an ICC has its secret keys compromised, thus preventing possible replay attacks.

**2. Limitations**

**a. Slow in Comparison to AES Only Authentication**

The AES encryption algorithm is currently used as the de facto standard for encryption in the smart card industry today [29]. Many smart card manufacturers opt to implement AES, as this form of encryption is computationally inexpensive and yet still relatively secure. This is understandable as smart cards have limited processing power and memory to perform complex encryption algorithms and adding additional memory and processors to handle such demands can come with steep price tags. As the PLAID protocol is required to perform *both* asymmetric encryption and symmetric encryption, its performance cannot match that of protocols using *only* symmetric encryption. In a study conducted by Sano et al on high performance smart cards using AES encryption, specifically Rijndael, with a 128 bit key length, it was determined that



it would take a total of 35,000 clock cycles on a 50Mhz microprocessor to encrypt the data [30]. This works out to be approximately 70ms, while data obtained from the PLAID protocol specification is approximately 200ms.

***b. Key Distribution Problem***

Since the PLAID protocol utilizes symmetric keys for encryption, there is still the problem pertaining to key distribution. Although the private keys are obtained from the key sets, vendors are still required to build key management for the PLAID protocol into existing or new key management systems. This may prove to be difficult to manage in the event that the protocol is implemented on a large scale.

Based on the guidelines found in the FIPS 140–2 publication, the security requirements for cryptographic key management include random number and key generation, key establishment, key distribution, key entry/output, key storage, and key zeroization. The PLAID protocol fulfills the majority of these requirements except in the area of key distribution where the implementation is often left to vendors. The PLAID protocol specifications do not provide any direction or guidance on such matters, as this aspect of the overall security system is outside the scope of the authentication protocol.

## **V. METHODOLOGY**

### **A. INTRODUCTION**

This chapter discusses both the conceptual framework and the practical elements of the research conducted. It explores the research question in more depth, and discusses the various methods used that are most appropriate given the aims and nature of the research. Various standards and guidelines provided the foundation for creating the list of factors used for comparing both protocols. Each of the factors selected from these standards and guidelines will be discussed, explaining why they were selected for the purposes of comparison.

### **B. DATA ANALYSIS**

Due to the nature of the research project where the result of data collection is qualitative in nature, it was deemed that the application of the Grounded Theory qualitative research approach through document analysis of the various security standards and guidelines for secure systems was the best research method to adopt. The utility of the Grounded Theory is a complex iterative process. It first begins with the raising of questions that help guide the research but are not intended to be confining. As data is gathered, core theoretical concepts are identified and linkages established between the concepts and data [31].

#### **1. Open Coding**

When applying Grounded Theory, the first process was the use of open coding to establish common broad themes from the collected data that was related to the research question. Here, all standards and guidelines used by contactless smart cards were reviewed so as to derive a broad list of factors that could potentially be used as a basis for comparison between the OPACITY and the PLAID protocols.

## 2. Axial Coding

Axial coding is the second pass through collected data, where the already identified themes from the open coding process are reanalyzed and filtered to produce a more detailed perspective whilst verifying the validity of the data. Through the use of this process, themes are delved deeper to create sub-themes resulting in the convergence of different factors obtained from the different standards and guidelines. The result of axial coding was the creation of a table with a detailed description of the various themes used for comparison. The result is summarized in Table 3:

**Table 3. Compilation of Reviewed Standards**

S/N	Document	Requirements	Description	Location
1	FIPS 140-2	Cryptographic Module Specification	Approved algorithms and approved modes of operation - Level 1-4	Section 4.1 Cryptographic Module Specification pg13-14
2	FIPS 140-2	Roles, Services and Authentication	<u>Roles</u> - Authorised roles for operators 1.) User Role 2.) Crypto Officer Role 3.) Maintenance Role - Level 1-4	Section 4.3 Roles, Services and Authentication Subsection 4.3.1 Roles pg.15-19
3	FIPS 140-2	Roles, Services and Authentication	<u>Services</u> Perform approved security functions - Level 1-4	Section 4.3 Roles, Services and Authentication Subsection 4.3.2 Services pg.15-19
4	FIPS 140-2	Roles, Services and Authentication	<u>Operator Authentication</u> Level 2 - Roles based or identity based authentication >= Level 3: Need Identity based operator authentication	Section 4.3 Roles, Services and Authentication Subsection 4.3.3 Operator Authentication pg.15-19
5	FIPS 140-2	Cryptographic Key Management	<u>RNG</u> - Deterministic VS Non-Deterministic RNGs - No approved non-deterministic RNGs exist - Level 1-4	Section 4.7 Cryptographic Key Management pg. 30-33
6	FIPS 140-2	Cryptographic Key Management	<u>Key Establishment</u> - Approved Key establishment - Level 1-4	Section 4.7 Cryptographic Key Management pg. 30-33
7	FIPS 140-2	Cryptographic Key Management	<u>Key Generation</u> - Based on approved key generation method - Level 1-4	Section 4.7 Cryptographic Key Management pg. 30-33

8	FIPS 201-2	Assurance Level for Physical Access Control	Local Workstation Some - CHUID / PKI-CAK High - Bio Very High - Bio-A / PKI-Auth Remote Network/Environment Some - PKI-CAK High - Very High - PKI-Auth	Section 6 PIV Cardholder Authentication pg.51-69
9	FIPS 201-2	Assurance Level for Logical Access Control	Logical Access Control Some - CHUID / ViS / PKI-CAK High - Bio Very High - Bio-A / PKI-Auth	Section 6 PIV Cardholder Authentication pg.51-69
10	FIPS 201-2	Relation between PIV and E-Authentication Assurance Level	Level 1 - Little Confidence PIV Level 2 - Some Level 3 - High Level 4 - Very High	Section 6 PIV Cardholder Authentication pg.51-69
11	SP 800-63-1	Tokens	1.) Number of factor of authentication a.) something you know b.) something you have c.) something you are  2.) Verifier generated token input (e.g. nonce or challenge) has at least 64 bits of entropy	Section 6 Tokens - pg. 40-54
12	SP 800-63-1	Credential Management	Strongly bound credentials (tamper evidence - digital signature) or weakly bound credentials.	Section 7 Token and Credential Management pg. 55- 67
13	SP 800-63-1	Credential Management	<u>Token and Credential Verification Services</u> Level 2 - Long term authentication secrets shall not be revealed to any party except verifiers + Cryptographic protections for private credentials for confidentiality and tamper protection + valid weak/strong bound credentials Level 3/4 - Secure Mechanisms to ensure that credentials are valid (temporary sessions keys) + Challenge and Response + Level 2	Section 7 Token and Credential Management pg. 55- 67
14	SP 800-63-1	Authentication Process	Resistance against (Picture Pg.77 - for Level) a.) Online guessing b.) Phishing and Pharming (verifier impersonation) c.) Eavesdropping d.) Replay resistance e.) Session Hijack f.) MITM  Level 2 - prove possession and control of token + Transmission confidentiality and integrity + Long term shared secret not revealed + Level 2 token + cookie for authentication Level 3 - Multifactor remote network authentication (level 3 token) Level 4 - Level 4 token	Section 8 Authentication Process pg. 67 - 80

15	SP 800-63-1	Assertion - make use of the idea	Holder of key or bearer of key assertion	Section 9 Assertion pg. 81-96
16	SP 800-78-3	Algorithm and Key length for identification	Recommendation of key length for identification information	Entire document
17	SP 800-131A	Cryptographic Algorithms and Key Lengths	Recommendation of validity of key lengths	Entire Document
18	NSA Website	Protection of Classification requirements	The types of cryptography and associated key length for protection of classified materials	website: <a href="http://www.nsa.gov/ia/programs/suiteb_cryptography/">http://www.nsa.gov/ia/programs/suiteb_cryptography/</a>
20	SP800-116	Threat Environment	Types of threats	Section 4 Threat Environment pg. 12-15
22	SP800-116	No of factors of authentication for different techniques	Similar to line item 11	Section 7 PIV Authentication Mechanisms Pg.24-31
23	SP800-116	Number of Factors of Authentication for different areas	Controlled - 1 Limited - 2 Exclusion - 3	Section 7 PIV Authentication Mechanisms Pg.24-31
24	SP800-116	Types of authentication techniques to controlled areas	Similar to line item 11	Section 7 PIV Authentication Mechanisms Pg.24-31

### 3. Selective Coding

The final pass in the coding process would be the use of selective coding. Here, the derived themes and codes from the previous two coding processes are combined and analyzed repeatedly to ensure that these themes and code provide a good basis for comparing the two different protocols while also answering the research question. The result of this coding process was the creation of a table that accurately identifies factors relating to this research project, as shown in Table 4:

**Table 4. Compilation of Reviewed Standards**

S/N	Comparison Factors	Description	References
1	Factors of Authentication	Examines the types and the number of factors of authentication employed, as well as the assurance and confidence level provided for physical and logical access	FIPS 140-2 Section 4.3 Roles, Services and Authentication, FIPS 201-2 Section 6 PIV Cardholder Authentication, SP800-63-1 Section 6 Tokens, SP800-63-1 Section 8 Authentication Process, SP800-116 Section 7 PIV Authentication Mechanisms
2	Granularity of Identity	Examines whether the authentication is performed at a fine granularity to achieve identity based authentication or at a coarse level to achieve role based authentication	FIPS 140-2 Section 4.3 Roles, Services and Authentication
3	Credential Confidence	Examines the assurance level on the integrity of the authentication data	SP800-63-1 Section 7 Token and Credential Management, SP800-63-1 Section 8 Authentication Process
4	Subject-Token Binding	Examines the extent of the binding between the subject and the token to prove if the cardholder is the rightful owner of the token	SP800-63-1 Section 7 Token and Credential Management, SP800-63-1 Section 8 Authentication Process, SP800-63-1 Section 9 Assertion.
5	Non-Repudiation	Examines whether a subject is able to deny participating in a transaction	SP800-63-1 Section 7 Token and Credential Management, SP800-63-1 Section 8 Authentication Process
6	MITM Resistance	Examines the resistance against MITM attack	SP800-63-1 Section 8 Authentication Process, SP800-116 Section 4 Threat Environment
7	Protection of Classified Material	Examines the suitability of the protocols to protect data of different classification levels	NSA Suite B Cryptography
8	Authorization Support	Examines the types of authorisation mechanisms supported by the protocols	NIST 7316 Assessment of Access Control System
9	Key Management	Examines if the complexity of the key distribution problem	General educational materials e.g. textbooks and research papers on key management
10	Bit Entropy	Examines the effective key space provided by various key and algorithm combinations	NSA The Case for Elliptic Curve Cryptography
11	Cipher Performance	Examines the latency of the protocols	General educational materials e.g. textbooks and research papers on speed of cryptography

The description of each of the comparison factors will be discussed when addressing the various standards and guidelines.

### **C. STANDARDS AND GUIDELINES**

Many standards and guidelines were reviewed to determine the most apropos to compare and contrast the protocols. One discerning issue was that many of these standards and guidelines are either too vague, containing only generic information regarding security requirements; or include technical coverage which is not applicable to answering the research question. An example to illustrate this is the Health Insurance Portability and Accountability Act (HIPPA), which stipulates generic security safeguards that must be implemented to control access to computer systems containing public health information, but does not specify the type of factors that must be employed when

doing so. Two other standards that were considered include EMV and ISO 14443. EMV is a global standard for inter-operation of ICCs and ATMs for authenticating credit and debit card transactions, while ISO 14443 is an international standard that defines contactless cards used for identification and the transmission protocols for communication. These standards provide comprehensive technical information relating to the physical implementation of the interactions between the smart cards and card terminal but there is limited information regarding the implementation of security mechanisms. These standards were thus not suitable for deriving comparison factors in our report.

#### **1. Federal Information Processing Standards 140–2 Publication (FIPS Pub 140–2)**

This standard specifies the security requirements that will be satisfied by a cryptographic module utilized within a security system. The publication helps coordinate the requirements and standards for cryptography modules that include both hardware and software components. Essentially, the publication defines four levels of security, from “Level 1” to “Level 4.” “Level 1” provides the lowest level of security. Basic security requirements are specified for a cryptographic module but no specific physical security mechanisms are required. “Level 2” improves upon the physical security mechanisms of “Level 1” by requiring features to have anti-tampering mechanisms that protect against unauthorized physical access. Physical security mechanisms required at “Level 3” are intended to have identity based authentication mechanisms, enhancing the security provided by the role based authentication mechanisms specified at “Level 2.” Finally, “Level 4” provides the highest level of security by providing holistic protection around the cryptographic module with the intent of both detecting and *responding* to all unauthorized attempts at physical access.

Specific sections of this publication are found to be of relevance when comparing the different protocols. Firstly, *Section 4.1* of the publication relates to the cryptographic module specification where it states that the cryptographic module shall implement at least one approved security function used in an

approved mode of operation. What we interpret from the term 'security function' would be how the protocols are required to use approved algorithms to perform encryption so as to protect the data they are sending. As mentioned previously in both Chapters III and IV, both protocols have many different modes of operations. Hence, we wanted to do a comparison that specifically looks at the factors of authentication, using approved algorithms, in a particular mode of operation.

Next, *Section 4.3* mentions the need to have authentication mechanisms within a cryptographic module to authenticate and verify that the operator accessing the module is authorized to perform the stipulated task. However, there are two different kinds of authentication mechanisms that may be implemented. Firstly, role-based authentication mechanisms authenticate only the set of roles stipulated and do not authenticate the individual identity of the operator. The second kind would be identity-based authentication where the module requires the operator to be individually identified before authorizing the operator to perform a task. We thought that by understanding the different type of authentication mechanisms found in the two protocols, it would provide a good basis for comparison.

Lastly, *Section 4.7* specifies the requirements for cryptographic key management that includes issues such as random number and key generation, key establishment and key distribution. Using the various sections in the publication as reference, three factors, namely, "**Factors of Authentication**," "**Granularity of Identity**" and "**Key Management**" were derived to compare both protocols. 'Factors of Authentication' examines the number of factors of authentication employed by both protocols, as well as the assurance and confidence level provided for both physical and logical access. The term 'Granularity of Identity' examines the level of granularity that the authentication mechanisms perform, which could range from identity-based authentication to role-based authentication. Finally, 'Key Management' examines the protocols for potential key distribution issues.



## 2. Federal Information Processing Standards 201–2 Publication (FIPS Pub 201–2)

This standard specifies the architecture and technical requirements for a common identification standard for Federal employees and contractors. The goal of this standard is to achieve the appropriate security assurance for multiple applications by verifying the claimed identity of these individuals seeking physical access to these government facilities and logical access to government information systems. In addition, the standard also provides detailed specifications that will support technical interoperability among personal identity verification (PIV) systems of Federal department and agencies. It describes the card elements, security controls and interfaces required to securely retrieve, process and store identity credentials from the card. This publication provides important factors for doing comparison between the two protocols such as the assurance level provided by different authentication mechanisms for both physical and logical access control.

**Table 5. Authentication for Physical Access**

<b>PIV Assurance Level Required by Application/Resource</b>	<b>Applicable PIV Authentication Mechanism</b>
SOME confidence	VIS, CHUID, PKI-CAK
HIGH confidence	BIO
VERY HIGH confidence	BIO-A, PKI-AUTH

**Table 6. Authentication for Logical Access**

<b>PIV Assurance Level Required by Application/Resource</b>	<b>Applicable PIV Authentication Mechanism</b>	
	<b>Local Workstation Environment</b>	<b>Remote/Network System Environment</b>
SOME confidence	CHUID, PKI-CAK	PKI-CAK
HIGH confidence	BIO	
VERY HIGH confidence	BIO-A, PKI-AUTH	PKI-AUTH

This standard also defines a suite of identity authentication mechanisms that are supported by all the PIV cards and their applicability in meeting the requirements for identity assurance. This section provided another means for comparing both protocols as their authentication mechanisms were very different. The PLAID protocol has PIV biometric authentication mechanisms that can be leveraged upon to provide better security. Hence, this section reinforces the notion of “**Factor of Authentication**” as it addresses the level of assurance and confidence of the authentication scheme.

### **3. NIST Special Publication 800–63–1: Electronic Authentication Guideline.**

The NIST special publication 800–63–1 provides technical guidelines to agencies that allow employees to remotely authenticate their identity to a Federal IT system using widely implemented methods for remote authentication such as PKI. With such methods, the individuals to be authenticated prove that they have possession of established secrets. The document also describes four assurance levels with qualitative degrees of confidence in the asserted identity’s validity; ranging from “Level 1” where there is little or no confidence, to “Level 4” where there is very high confidence. In particular, the sections we focused on addressed the technical requirements for each of the four levels of assurance, specifically in the areas of token and credential management mechanisms used to establish and maintain token and credential information, as well as the protocols used to support the authentication mechanisms found in *Section 7 and Section 8* of the publication. Table 7 provides an excellent representation of the assurance levels for multi-token authentication schemes:

**Table 7. Assurance Levels for Multi-Token Authentication Schemes**

	Memorized Secret Token	Pre-registered Knowledge Token	Look-up Secret Token	Out of Band Token	SF OTP Device	SF Cryptographic Device	MF Software Cryptographic Token	MF OTP Device	MF Cryptographic Device
Memorized Secret Token	Level 2	Level 2	Level 3	Level 3	Level 3	Level 3	Level 3	Level 4	Level 4
Pre-registered Knowledge Token	X	Level 2	Level 3	Level 3	Level 3	Level 3	Level 3	Level 4	Level 4
Look-up Secret Token	X	X	Level 2	Level 2	Level 2	Level 2	Level 3	Level 4	Level 4
Out of Band Token	X	X	X	Level 2	Level 2	Level 2	Level 3	Level 4	Level 4
SF OTP Device	X	X	X	X	Level 2	Level 2	Level 3	Level 4	Level 4
SF Cryptographic Device	X	X	X	X	X	Level 2	Level 3	Level 4	Level 4
MF Software Cryptographic Token	X	X	X	X	X	X	Level 3	Level 4	Level 4
MF OTP Device	X	X	X	X	X	X	X	Level 4	Level 4
MF Cryptographic Device	X	X	X	X	X	X	X	X	Level 4

The respective sections provided additional factors such as “**Credential Confidence**,” “**Subject Token Binding**,” “**Non Repudiation**” and “**Man-in-the-middle (MITM) Resistance**.” Here, the term ‘Credential Confidence’ refers to the amount of assurance the relying party (i.e., the entity that is authenticating a subject as a prerequisite to granting some access) has over the integrity of the secret being used to authenticate a subject. Next, the term “Subject Token Binding” examines the extent of the binding between the subject and the token so as to prove that the subject is the rightful owner of the token and ‘Non-Repudiation’ examines if the subject is able to deny having participated in a particular transaction. Lastly, the protocols are examined to assess their resistance to MITM attacks.

4. **NIST Special Publication 800–116: A Recommendation for the Use of PIV Credentials in Physical Access Control Systems (PACS).**

This document describes a strategy for selecting the appropriate PIV authentication mechanisms to manage physical access to government facilities and assets. In addition, this document also describes the desired characteristics of a target implementation of PIV enabled PACS. It also discusses some of the PIV card capabilities so a that risk-based assessment can be aligned with the appropriate PIV authentication mechanism. However, the information that was most relevant is in *Section 7* where recommendations are provided on the use of PIV authentication mechanisms in a PACS environment. Table 8 illustrates the many different combinations of authentication mechanisms and the number of factors required.

**Table 8. Authentication Factors of PIV Authentication Mechanisms**

PIV Authentication Mechanism	Have	Know	Are	Authentication Factors (HKA Vector)	Interface
CAK + BIO (-A)	x	x	x	3	Contact
BIO-A	x		x	2	Contact
PKI	x	x		2	Contact
BIO			x	1	Contact
CAK	x			1	Contact/ Contactless
CHUID + VIS	x			1	Contact/ Contactless

The publication also provides recommendations on how authentication mechanisms ought to be utilized based on the protective areas established around assets or resources. Using the concept of “Controlled, Limited or Exclusion” areas, the number of authentication factors that is recommended is shown in Table 9.

**Table 9. Authentication Factors for Security Areas**

Security Areas	Number of Authentication Factors Required
Controlled	1
Limited	2
Exclusion	3

The points brought up in these sections reinforced the concept of having **“Factors of Authentication”** as a means of comparison of the two different protocols.

#### **5. NSA Suite B Cryptography website / NSA The Case for Elliptic Curve Cryptography**

To ensure that our research was thorough and rigorous, we also looked at some of the recommendations made by the NIST publication 800–131A addressing issues pertaining to the cryptographic algorithms and key lengths. In addition, we also looked up the National Security Agency (NSA) website to determine the type of cryptography used to protect national security systems and national security information. The Suite B cryptography utilizes AES with 128-bit keys to provide protection for classified information up to the SECRET level. Similarly, an ECDSA using a 256-bit prime modulus elliptic curve also provides adequate protection for classified information up to SECRET level [32]. However, to protect information classified as TOP SECRET, there is a need to use either AES with 256-bit keys or 384-bit prime modulus ECC. Another NSA article talks about the apparent benefits of using elliptic curve cryptography [33]., The article addresses the issue of bit entropy and how the use of ECC provides much better security as a result of having higher bit entropy.

After reviewing the respective materials, we felt that factors such as **“Protection of Classified Material”** and **“Bit Entropy”** were important attributes that could be used to benchmark the performance of the two different protocols. Specifically, the ‘Protection of Classified Material’ examines the suitability of the

protocols to protect data of different classification levels, while ‘Bit Entropy’ examines the per bit security provided by both the OPACITY and the PLAID protocols.

## **6. Payment Card Industry Data Security Standard (PCI-DSS) / ISO 7816–8**

Two other standards that were reviewed were the PCI-DSS and ISO 7816–8. The PCI-DSS was developed to encourage and enhance cardholder data security by providing a baseline of technical and operational requirements designed to protect cardholder data. Some of the requirements stipulated in the standard included the encryption of cardholder data across public networks and the assignment of a unique ID to users. Many of these requirements were vague and thus not deemed suitable factors for comparison. Similarly, when reviewing the ISO 7816–8 standard, a standard that specifies inter-industry commands for integrated circuit cards that may be used for cryptographic operations, the requirements were too technical and were not useful in helping us derive additional factors for comparison.

Of all the standards and guidelines that were examined, none of them focuses on the *performance* of various cryptographic mechanisms. However, such performance directly influences the operational acceptability (speed of transaction) of any given mechanism and is thus an important factor of consideration when deciding which cryptographic mechanisms to adopt for real time operations. Hence, “Cipher Performance,” which examines the latency of cryptographic algorithms, is another factor identified to compare the protocols.

THIS PAGE INTENTIONALLY LEFT BLANK

## **VI. COMPARISONS AND FINDINGS**

### **A. Comparisons**

The OPACITY and the PLAID protocols are privacy-enhanced protocols that perform mutual authentication, provide protection of information that is exchanged between the smart card and the card terminal to protect against most types of attacks on smart card systems. The protocols use a mixture of symmetric and asymmetric cryptography during the authentication process. The cryptography is based on ISO, FIPS or NIST approved or recommended cryptographic algorithms and modes of operation, such as ECDH, RSA 256 and AES 256. In addition to authentication, the protocols also provide support for authorization processes.

Despite the similarities between the protocols, there are distinct differences between the protocols. This chapter uses the list of factors identified in Chapter V to compare the protocols. A summary of the comparison is presented in Table 10.



**Table 10. Summary of Comparisons**

S/N	Comparison Factors	OPACITY	PLAID
1	Factors of Authentication	Single factor authentication based on possession of smart card	Able to support up to three factors authentication based on possession of smart card, PIN authentication and biometric authentication.
2	Granularity of Identity	Identity-based operator authentication	Role-based operator authentication or identity-based operator authentication
3	Credential Confidence	Strong assurance on the integrity of the credentials	Weak assurance on the integrity of the credentials
4	Subject-Token Binding	Weak binding between subject and token	Able to achieve strong binding between subject and token
5	Non-Repudiation	Does not provide user non repudiation.	Able to provide user non-repudiation
6	MITM Resistance	Vulnerable to relay attack	Able to provide resistance to relay attack
7	Protection of Classified Material (Classification Level)	Suitable to be used for protection of classified data up to Top Secret level.	Suitable to be used for protection of classified data up to Secret level.
8	Authorization Support	Able to support object oriented authorisation process	Able to support object oriented and subject oriented authorisation process
9	Key Management	Minimise key distribution problem.	Potential key distribution problem
10	Bit Entropy	Higher bit entropy	Lower bit entropy
10	Cipher Performance	Slower performance but may become faster than PLAID protocol if PLAID protocol requires RSA operations with key length greater than 2048 bits	Faster performance but may become slower than OPACITY protocol if the protocol requires RSA operations with key length greater than 2048 bits

## 1. Factors of Authentication

The OPACITY protocol is a single factor authentication protocol, while the PLAID protocol is able to be either a single or multi-factor authentication protocol, employing up to three factors of authentication.

The OPACITY protocol's single factor authentication is based on proof of possession of a valid smart card. The single factor authentication protocol does not achieve a high assurance level for the token or the authentication scheme for electronic authentication [28]. The protocol is only able to achieve, at most, level 2 assurance for the token. This means that the protocol can only achieve, at most, level 2 assurance for the *overall* authentication scheme. For personal identification verification, the protocol only provides *some* confidence in the asserted identity for local and remote physical access requests, as well as for logical access requests [34]. If the protocol is to be used for physical access, the protocol is only recommended to be used for authentication to controlled areas [27].

The PLAID protocol supports multi-factor authentication by employing PIN authentication or biometric authentication based on a biometrically unique feature, in addition to the proof of possession of a valid smart card. The protocol can be used as a two-factor authentication protocol by employing either PIN authentication or biometric authentication in addition to the proof of possession of a valid smart card; and it can be used as a three-factor authentication protocol by employing PIN authentication and biometric authentication, as well as proof of possession of a valid smart card. With multi-factor authentication, the protocol is able to achieve up to level 3 assurance for the token and *overall* authentication scheme for electronic authentication [28]. For personal identification verification purposes, the multi-factor authentication protocol is able to provide *very high* confidence in the asserted identity for local and remote physical access requests, as well as logical access requests [34]. If the protocol is to be used for physical access, the protocol is suitable to be used for authentication to controlled, limited and exclusive areas [27].

## **2. Granularity of Identification**

The OPACITY protocol performs identity-based authentication, while the PLAID protocol performs role-based authentication.

Both of the protocols performs authentication based on PKC. PKC is able to perform identity-based authentication or role-based authentication, depending on the implementation of the private keys. If the private keys are uniquely assigned to each claimant, identity based authentication is achieved. On the other hand, if the private keys are shared among a number of claimants, role-based authentication is achieved. The recommended implementation is that the private keys shall be uniquely assigned to each claimant to achieve identity-based authentication to verify the identity of the claimant and the authorization of the claimant to assume the selected role [35].

The OPACITY protocol follows the recommendation to perform identity-based authentication. For the PLAID protocol, authentication is performed at the building, role or function level to achieve role-based operator authentication. Each building, role or function is assigned a KeySetID and a pair of RSA keys. The KeySetID and the corresponding public RSA key is stored at the card terminal while the KeySetID and the corresponding private RSA key is stored on the smart card. During authentication, the protocol verifies that the smart card and the card terminal have the correct KeySetID and the corresponding key pairs. However, the protocol is able to achieve identity-based operator authentication if biometric authentication is also employed. As the biometric feature is unique to each individual, the biometric feature can be used to uniquely identify each individual to achieve identity-based authentication

## **3. Credential Confidence**

The OPACITY protocol provides tamper-evident protection and strong assurance on the integrity of the credentials while the PLAID protocol does not.

The OPACITY protocol performs authentication based on PKC. The CVC, which contains the credentials (i.e., identification information and user's public

authentication key), is defined based on the X.509 format for public certificate and is digitally signed using the root private digital signature keys of the issuer. The card terminal's CVC is digitally signed using the root private digital signature key for the card terminal domain while the smart card's CVC is digitally signed using the root private digital signature for the smart card domain. During authentication, the digital signatures are verified using the corresponding root public digital signature keys from the respective domains before the credentials in the CVCs are used. The use of digital signature provides tamper-evident protection and strong assurance on the integrity of the credentials in the CVC.

The PLAID protocol does not provide tamper-evident protection and assumes that the integrity of the credentials is preserved. The ACS record corresponding to the requested OpModelID, is returned from the smart card to the card terminal and used for authorization without any verification of the integrity of the record. This does not provide strong assurance as the ACS record can be injected or modified by an attacker who may be a legitimate user without the authorization rights to gain unauthorized access to the requested resource.

#### **4. Subject-Token Binding**

The OPACITY protocol exhibits weak binding between the subject and the token, while the PLAID protocol is able to achieve strong binding between the subject and the token.

The OPACITY protocol exhibits weak binding between the subject and the token as the protocol does not require the claimant to prove control of the smart card. In addition, as all the required authentication information is stored in the smart card, the claimant will be successfully authenticated, even if the claimant is not the rightful owner of the smart card, as long as the claimant possesses a valid smart card. Without proving control of the smart card during the authentication process, the protocol does not even achieve the minimum assurance level for the authentication scheme for electronic authentication [28].

The PLAID protocol achieves strong binding between the subject and the token by employing multi-factor authentication, such that the claimant needs to prove possession and control of the smart card during the authentication process. Proof of control of the smart card is achieved by employing PIN authentication, biometric authentication, or both. PIN authentication is performed by requiring the claimant to enter a PIN at the card terminal and comparing the hash of the entered PIN to a copy of the hash of the PIN that is stored in the smart card. Since the PIN should only be known by the rightful owner, the claimant would have successfully proved control/ownership of the smart card if the hashes match. Biometric authentication is performed by scanning a biometrically unique feature of the claimant and comparing the biometrics to the minutiae that is stored in the smart card. As the biometric is unique to the claimant, the claimant would have successfully proved control/ownership of the smart card if the biometrics match. If the protocol does not employ either PIN authentication or biometric authentication, the protocol exhibits weak binding between the human subject and the token, as is the case with OPACITY protocol.

## **5. Non-Repudiation**

The OPACITY protocol does not achieve user non-repudiation, while the PLAID protocol does.

The OPACITY protocol does not achieve user non-repudiation due to the weak binding between the subject and the token. The owner could claim that he had lost his smart card, and the smart card was picked up by an unknown party to make the transaction. The protocol only achieves “non-repudiation” in that a particular smart *card* is used in the transaction; vice a particular person.

The PLAID protocol achieves user non-repudiation by employing biometric authentication based on biometrically unique feature, in addition to proof of possession of a valid smart card. As the biometric feature uniquely identifies the claimant, the claimant cannot deny participation in the transaction. If the protocol

does not employ biometric authentication, the protocol does not achieve user non-repudiation, as is the case with the OPACITY protocol.

## **6. Man-in-the-Middle Resistance**

Both of the protocols are resistant to MITM attacks such as active eavesdropping and modification attacks. However, the OPACITY protocol is vulnerable to relay attack while the PLAID protocol is not.

The OPACITY protocol is vulnerable to relay attack because the protocol only requires the claimant to prove possession of the smart card and does not require the claimant to prove control/ownership of the smart card. Using a relay attack, the card terminal is tricked into believing that the claimant possesses a valid smart.

The PLAID protocol provides resistance against relay attacks if the protocol employs PIN or biometric authentication in addition to proof of possession of a valid smart card. In this case, the attacker will not be able to produce the PIN or biometric feature for authentication, mitigating the relay attack. If the protocol does not employ PIN or biometric authentication, the protocol becomes vulnerable to relay attack.

## **7. Protection of Classified Material**

The OPACITY protocol can be used to protect classified material up to the top-secret level, while the PLAID protocol can only be used for protection of classified material to the secret level.

During the authentication process, the protocols derive session keys that are used to protect the data that is exchanged between the smart card and the card terminal after successful authentication. These keys are either used for encryption or hashing. The OPACITY protocol is able to derive different session keys for encryption and hashing while the PLAID protocol is only able to derive one session key for both operations. In addition, for the PLAID protocol the

length of the session key needs to be similar to the size of the AES cipher selected, which implies that the maximum key length can only be up to 256 bits.

To protect top-secret data, the NSA requires an AES with 256-bit keys, Elliptic Curve Public Key Cryptography using the 384-bit prime modulus, and SHA with 384-bit keys [32]. The OPACITY protocol is configurable to support the required key lengths to provide protection of data up to the top-secret level. However, for the PLAID protocol the maximum key length of 256 bits is insufficient to support SHA-384 operation. The maximum key length of 256 bits is only sufficient to provide protection up to the secret level.

## **8. Authorization Support**

Both of the protocols are able to support authorization processes. However, the OPACITY protocol is only able to support object-oriented authorization, while the PLAID protocol is able to support object-oriented authorization and subject-oriented authorization [36].

For the OPACITY protocol, the identification information in the CVC can be verified by comparing it against an access control list (object-oriented), which is stored centrally at the backend or locally at the card terminal; to determine if the claimant is authorized access to the requested resource.

For the PLAID protocol, the smart card returns the ACS record corresponding to the requested OpModelID to the card terminal for authorization. The identification information in the ACS record can be verified by comparing it against an access control list (object-oriented) in a similar manner to the OPACITY protocol. Alternatively, the ACS record can be used as a capability list (subject-oriented). When the card terminal polls for the ACS record corresponding to the OpModelID, the presence of an ACS record corresponding to that OpModelID is an indication of whether the claimant is authorized access to the requested resource. Only if a record is present is the claimant authorized access to the requested resource.

## **9. Key Management**

The OPACITY protocol has minimized key distribution problems, while the PLAID protocol has a potential key distribution problem.

The OPACITY protocol has minimized key distribution problems because the protocol is based on PKC. One advantage of PKC is that it has minimized key distribution problems, as there is no need for pre-defined shared secrets to be securely distributed to all of the parties that may utilize them. Instead, all the shared secrets are derived and established during the authentication process using ECDH or a predefined concatenation of KDFs. However, CVC revocation functionality is not designed into the protocol. In the event that a smart card is compromised and there is no CVC revocation functionality, the user identification information and associated authorization privileges, in addition to the keys, need to be removed from the backend system, re-generated, re-issued, and updated to the backend system. This requires careful key management from the backend system.

The PLAID protocol has a potential key distribution problem because of the need for predefined shared secrets. The key distribution problem is further aggravated by the application of the RSA keys in a symmetric manner. Each building, role or function is assigned a KeySetID and a pair of RSA keys. The KeySetID and the public RSA key are stored at the card terminal, while the KeySetID and the private RSA key are stored on the smart card. Users who access the same building, have the same role, or perform the same function will share the same respective private RSA keys. In the event that a smart card is compromised (i.e., the keys are recovered by an attacker), the protocol relies on key diversification to ensure that the system remains secure [5]. However, if the attacker is able to recover the keys stored on the smart card, it is likely that the attacker is also able to recover the diversification data stored on the smart card. Hence, key diversification does not provide effective protection to ensure that the system remains secure if the smart card is compromised. A better approach is to rely on multi-factor authentication based on PIN or biometric authentication.



Since the attacker is unlikely to know the PIN or possess the same biometric feature, the system remains secure. However, it is possible that an attacker obtains possession of a valid smart card and manages to alter the PIN hash and minutiae stored on the smart card to achieve successful authentication. Therefore, the best practice is to revoke, replace and re-distribute all the compromised keys to all affected users and systems whenever a smart card is compromised. This results in a difficult key distribution problem in the long run.

## 10. Bit Entropy

The OPACITY protocol is able to achieve higher key strength per bit compared to the PLAID protocol.

The OPACITY protocol uses ECC while the PLAID protocol uses RSA. The main advantage of ECC over RSA is that ECC is able to achieve higher key strength per bit than RSA for the same key length. For instance, to achieve the same bit entropy of 128 bits for a symmetric key, ECC only requires a 256 bits key while RSA requires a 3072 bits key as shown in Table 11 [33]. This implies that to achieve a given security level, as measured by resistance to brute-force key guessing, the OPACITY protocol can achieve this level with a smaller, more efficient, key size as compared to the PLAID protocol.

**Table 11. NIST Recommended Key Sizes**

<b>Symmetric Key Size (bits)</b>	<b>RSA and Diffie-Hellman Key Size (bits)</b>	<b>Elliptic Curve Key Size (bits)</b>
80	1024	160
112	2048	224
128	3072	256
192	7680	384
256	15360	521

## 11. Cipher Performance

The OPACITY protocol is slower in performance compared to the PLAID protocol even if the PLAID protocol employs RSA up to a key length of 2048 bits. This is because the PLAID protocol consists mainly of symmetric key operations

and only two asymmetric key operations (i.e. RSA encryption and decryption) while the OPACITY protocol consists of more symmetric and asymmetric operations. Even though the OPACITY protocol employs ECC, ECC only achieves faster performance in specific operations such as key generation and digital signature signing, which are not employed in the PLAID protocol [37], [38], [39].

However, the speed of RSA encryption and decryption operations decreases multiplicatively for a given increase in key length [40]. Hence, in the long run when RSA requires a much longer key length than 2048 bits to preserve a defined security level [41], the OPACITY protocol will make up for some of its speed disadvantage against the PLAID protocol.

## **B. SUMMARY OF FINDINGS**

The key advantage of the PLAID protocol over the OPACITY protocol is its capability to support multi-factor authentication. By employing multi-factor authentication, the PLAID protocol produces a stronger authentication scheme and provides higher assurance than the OPACITY protocol. As a result, the PLAID protocol is able to achieve a strong binding between the subject and the token and thus provide protection against relay attack and user repudiation. However, if multi-factor authentication is not employed, the PLAID protocol becomes a single factor authentication protocol, like the OPACITY protocol. In this case, the only advantages of the PLAID protocol over the OPACITY protocol are that the PLAID protocol provides more flexibility in authorization implementation and faster cipher performance if the RSA key length is shorter than 2048 bits.

On the other hand, the key advantage of the OPACITY protocol over the PLAID protocol is that it has simpler key distribution problem. This is attributed to the use of PKC in the OPACITY protocol. Other advantages include higher bit entropy and higher credential confidence. The cipher performance of the

OPACITY protocol will also become better than the PLAID protocol when RSA requires a much longer key length than 2048 bits to provide a given cryptographic security level.

Overall, the PLAID protocol is able to produce a stronger authentication scheme, it provides higher assurance and it is more versatile in implementation than the OPACITY protocol. The PLAID protocol, employing multi-factor authentication, is more suitable for use in higher risk environments than the OPACITY protocol. However, due to the more complex key distribution problem of the PLAID protocol, the PLAID protocol is not recommended for large-scale deployment. On the other hand, the OPACITY protocol has the simpler key distribution problem. However, for the OPACITY protocol to be used in high-risk environments, the protocol needs to be complemented with external security mechanisms. For instance, the backend will be responsible to perform further authentication such as PIN authentication. The claimant will be required to enter a PIN at the keypad and the entered PIN will be verified against a copy of the PIN that is stored in the backend. If the PINs match, authentication is successful. Otherwise, authentication is unsuccessful.

## VII. CONCLUSIONS

There is a proliferating use of contactless smart card systems for identification, financial transactions, and access control due to their convenience, performance, and the basic security that their employment provides, as well as the ease of integrating the systems for use in a wide range of other applications. However, despite their benefits, contactless smart card systems are vulnerable to attacks such as eavesdropping, modification, and MITM attacks.

The OPACITY protocol and the PLAID protocol are able to provide protections against most of these attacks and ensure secure communications between the smart card and the card terminal. Analysis of the protocols revealed that the PLAID protocol is able to produce an overall stronger authentication scheme and provide higher assurance when multi-factor authentication is employed. If the PLAID protocol does not employ multi-factor authentication, it will only provide equivalent assurance to the OPACITY protocol. The higher assurance provided by a PLAID protocol that employs multi-factor authentication makes the protocol suitable to be used in higher risk environments than the OPACITY protocol. However, due to the more complex key distribution problem of the PLAID protocol, it is not recommended for large-scale deployment. In those cases, it is better to use the OPACITY protocol, which has a simpler key distribution problem. However, for the OPACITY protocol to be used in high-risk environments, the protocol needs to be complemented with external security mechanisms to achieve multi-factor authentication.

As alluded to in Chapter I, Section C titled “Research Method,” this analysis was performed qualitatively using materials from open literature. However, due to the limited literature on the OPACITY and PLAID protocols, the understanding of the OPACITY protocol and the PLAID protocol was derived by analyzing the protocols’ specifications and relevant documents on cryptography and authentication. As such, the analysis may not capture working details concerning these protocols. Hence, a next step of this research would be to

acquire the protocols and analyze their implementation in controlled laboratory experiments in order to measure their actual performance and perhaps identify additional areas that may affect the determination of which protocol is best for particular uses and environments.

## LIST OF REFERENCES

- [1] S. Posthumus and R. von Solms, "A framework for the governance of information security," *Computer & Security*, vol. 23, pp. 638–648, 2004.
- [2] M. Gerber and R. von Solms, "From risk analysis to security requirements," *Computer & Security*, vol. 20, pp. 577–584, 2001.
- [3] Smart Card Alliance, "Contactless payment and the retail point of sale: applications, technologies and transaction models," March 2003. [Online].
- [4] B. Walder, "Smart cards - the use of 'intelligent plastic' for access control," [Online]. Available:
- [5] Smart Card Alliance Identity Council, "RF-enabled applications and technology: comparing and contrasting rfid and rf-enabled smart cards,"
- [6] P. Oswal and M. Foong, "RFID vs contactless smart cards - an unending debate," Frost & Sullivan, 4 October 2006. [Online]. Available:
- [7] Frost & Sullivan, "World of contactless smart card market," Frost & Sullivan, 18 Feb 2008.
- [8] Frost & Sullivan, "Asia-pacific smart card integrated circuit on different form factors," Frost & Sullivan, 27 Aug 2010.
- [9] IMS Research, "World market for smart cards and smart card ics - 2011 Edition," IMS Research, 2011.
- [10] U.S. Department of Health and Human Services, "Health information privacy," [Online]. Available: <http://www.hhs.gov/ocr/privacy/hipaa/>
- [11] Smart Card Alliance, "Alliance activities : publications : transportation," [Online]. Available:
- [12] Smart Card Alliance, "About smart cards : applications : emv," [Online]. Available: <http://www.smartcardalliance.org/pages/smart-cards->
- [13] Smart Card Alliance, "About smart cards : frequently asked questions," [Online]. Available: <http://www.smartcardalliance.org/pages/smart-cards->
- [14] D. H. Handschuh, "Contactless technology security issues," *Information Security Bulletin*, vol. 9, pp. 95–100, April 2004.
- [15] K. Ziv and W. Avishai, "Picking virtual pockets using relay attacks on contactless smartcard systems," in *Conference on Security and Privacy*
- [16] L. Francis, K. M. Gerhard Hancke and K. Markantonakis, "Practical relay attack on contactless transactions using nfc mobile phones," *Cryptography*
- [17] E. Le Saint, D. Fedronic and L. Steven, "Open protocol for access control identification and ticketing with privacy - specifications v3.7," 15 July 2011.
- [18] National Institute of Standards and Technology (NIST), "Security requirements for cryptographic modules," 25 May 2001. [Online].
- [19] National Institute of Standards and Technology (NIST), "Recommendation for pair-wise key establishment schemes using discrete logarithm

- [20] National Institute of Standards and Technology (NIST), "Recommendation for key management: part 1: general (revision 3)," July 2012. [Online].
- [21] ISO/IEC, "ISO/IEC 24727-4: identification cards -- integrated circuit card programming interfaces -- part 4: application programming interface
- [22] ISO/IEC, "ISO/IEC 7816-4: identification cards -- integrated circuit cards -- part 4: organization, security and commands for interchange," ISO/IEC,
- [23] C. Nita-Rotaru, "Lecture 20: key establishment, ipsec," 2005. [Online]. Available: <http://homes.cerias.purdue.edu/>
- [24] C. Kaufman, R. Perlman and M. Speciner, *Network Security Private Communication in a Public World*, Prentice Hall PTR, 2002.
- [25] National Institute of Standards and Technology (NIST), "A recommendation for the use of piv credentials in physical access control
- [26] National Institute of Standards and Technology (NIST), "Electronic authentication guideline," December 2011. [Online]. Available:
- [27] Centrelink, "Protocol for lightweight authentication of identity," December 2009. [Online]. Available:
- [28] L. C. Feng, "Fast implementation of aes cryptographic algorithms in smart cards," in *IEEE 37th Annual International Carnahan Conference on*
- [29] F. Sano, M. Koike, S. Kawamura and M. Shiba, "Performance evaluation of aes finalists on the high-end smart card," in *AES Candidate*
- [30] W. L. Neuman, *Basics of Social Research: Qualitative and Quantitative Approaches*, Pearson/Allyn and Bacon, 2006.
- [31] National Institute of Standards and Technology (NIST), "DRAFT personal identity verification of federal employees and contractors (revised draft),"
- [32] National Institute of Standards and Technology (NIST), "Entity authentication using public key cryptography," 18 February 1997. [Online].
- [33] National Security Agency (NSA), "NSA suite b cryptography," [Online]. Available: [http://www.nsa.gov/ia/programs/suiteb\\_cryptography/](http://www.nsa.gov/ia/programs/suiteb_cryptography/).
- [34] National Institute of Standards and Technology (NIST), "Assessment of access control system," [Online]. Available:
- [35] National Security Agency (NSA), "The case for elliptic curve cryptography," [Online]. Available:
- [36] H. Khurana, R. Koleva and J. Basney, "Performance of cryptographic protocols for high-performance, high-bandwidth and high-latency grid
- [37] N. Jansma and B. Arrendondo, "Performance comparison of elliptic curve and rsa," [Online]. Available:
- [38] RSA Laboratories, "Overview of elliptic curve cryptosystems," [Online]. Available: <http://www.rsa.com/rsalabs/node.asp?id=2013>. [Accessed 14
- [39] JAVAMEX, "Rsa key lengths," [Online]. Available: [http://www.javamex.com/tutorials/cryptography/rsa\\_key\\_length.shtml](http://www.javamex.com/tutorials/cryptography/rsa_key_length.shtml).

- [40] RSA Laboratories, "Twirl and rsa key size," [Online]. Available: <http://www.rsa.com/rsalabs/node.asp?id=2004#nist03>. [Accessed 2 June]
- [41] Actividentity CTO Office, "The open protocol for access control identification and ticketing with privacy for the secure enablement of
- [42] M. Gerber and R. von Solms, "From risk analysis to security requirements," *Computer and Security*, vol. 20, pp. 577–584, 2001.



THIS PAGE INTENTIONALLY LEFT BLANK

## INITIAL DISTRIBUTION LIST

1. Dudley Knox Library  
Naval Postgraduate School (NPS)  
Monterey, California
2. Chairman, Computer Science Department  
Naval Postgraduate School (NPS)  
Monterey, California
3. John D. Fulp  
Naval Postgraduate School (NPS)  
Monterey, California
4. Dr. Gurminder Singh  
Naval Postgraduate School (NPS)  
Monterey California
5. Jonathan Lee  
Defense Manpower Data Center (DMDC)  
Monterey, California
6. Dr. Yeo Tat Soon  
Director TDSI  
Temasek Defence Systems Institute (TDSI)  
National University of Singapore (NUS)  
Singapore
7. COL Lim Soon Chia  
Dy CRTO  
Defence Research and Technology Office (DRTech)  
Singapore Armed Forces (SAF)  
Singapore
8. Teo Tiat Leng  
Dy Dir LS  
Defence Science and Technology Agency (DSTA)  
Singapore
9. Tan Lai Poh  
Senior Manager TDSI  
Temasek Defence Systems Institute (TDSI)  
National University of Singapore (NUS)  
Singapore

10. Koh Ho Kiat  
Defence Science and Technology Agency (DSTA)  
Singapore
11. CPT Lee Yong Run  
Singapore Armed Forces (SAF)  
Singapore